
Computational Methods for Multi-Loop
Scattering Amplitudes with many
Kinematic Scales

DISSERTATION BY
Anton Olsson

to obtain the academic degree of
DOKTOR DER NATURWISSENSCHAFTEN (DR. RER. NAT.)

accepted by the KIT-Department of Physics of
Karlsruhe Institute of Technology (KIT)

Reviewer: Prof. Dr. Gudrun Heinrich

Second reviewer: Prof. Dr. Matthias Steinhauser

Date of Defence: 31 October 2025



This document is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0):

<https://creativecommons.org/licenses/by-sa/4.0/deed.en>

Eidesstattliche Versicherung gemäß § 13 Absatz 2 Ziffer 3 der Promotionsordnung des Karlsruher Instituts für Technologie (KIT) für die KIT-Fakultät für Physik:

1. Bei der eingereichten Dissertation zu dem Thema

**Computational Methods for Multi-Loop Scattering Amplitudes
with many Kinematic Scales**

handelt es sich um meine eigenständig erbrachte Leistung.

2. Ich habe nur die angegebenen Quellen und Hilfsmittel benutzt und mich keiner unzulässigen Hilfe Dritter bedient. Insbesondere habe ich wörtlich oder sinngemäß aus anderen Werken übernommene Inhalte als solche kenntlich gemacht.
3. Die Arbeit oder Teile davon habe ich bislang nicht an einer Hochschule des In- oder Auslands als Bestandteil einer Prüfungs- oder Qualifikationsleistung vorgelegt.
4. Die Richtigkeit der vorstehenden Erklärungen bestätige ich.
5. Die Bedeutung der eidesstattlichen Versicherung und die strafrechtlichen Folgen einer unrichtigen oder unvollständigen eidesstattlichen Versicherung sind mir bekannt.

Ich versichere an Eides statt, dass ich nach bestem Wissen die reine Wahrheit erklärt und nichts verschwiegen habe.

Karlsruhe, den 6. November 2025

.....
(Anton Olsson)

List of Publications

Articles

- S. P. Jones, A. Olsson, T. Stone. **JHEP** **10** (2025) 068, *Positive Integrands from Feynman Integrals in the Minkowski Regime*.
- V. Bresó, G. Heinrich, V. Magerya, and A. Olsson, “Interpolating amplitudes”, [arXiv:2412.09534](#), (Accepted for publication in *SciPost*).
- R. V. Harlander, T. Nellopoulos, A. Olsson, M. Wesle. **Comput. Phys. Commun.** **306** (2025), *ftint: Calculating gradient-flow integrals with pySecDec*.
- B. Agarwal, G. Heinrich, S. P. Jones, M. Kerner, S. Y. Klein, J. Lang, V. Magerya, A. Olsson. **JHEP** **05** (2024) 013, *Two-loop amplitudes for $t\bar{t}H$ production: the quark-initiated N_f -part*.
- G. Heinrich, S. P. Jones, M. Kerner, V. Magerya, A. Olsson, J. Schlenk. **Comput. Phys. Commun.** **295** (2024). *Numerical scattering amplitudes with pySecDec*.

Proceedings

- S. P. Jones, A. Olsson, T. Stone. **PoS LL2024** (2024) 036, *Evaluating Parametric Integrals in the Minkowski Regime without Contour Deformation*.

Abstract

Over the last few decades, in particular since the discovery of the Higgs boson in 2012, there has been a growing demand for accurate Standard Model predictions of high-energy scattering processes taking place at particle colliders. Experiments have dramatically improved the precision with which production rates and particle properties are being measured. In order to extract as much information from these measurements as possible, the precision of theoretical predictions must be brought to the same standard. A dominant source of theoretical uncertainty is due to missing higher-order terms in the perturbative expansion of scattering amplitudes. This thesis addresses this problem by introducing two novel techniques for the computation of multi-loop scattering amplitudes, with a focus on numerical methods for processes that involve many kinematic scales. The first method is a new procedure for resolving integrable singularities of Feynman loop integrals, by decomposing them into a sum of integrals over manifestly positive integrands, multiplied by complex prefactors. The resolved integrals can be integrated in the physical region without contour deformation, which accelerates numerical evaluation by up to several orders of magnitude. The second method is a framework for constructing interpolation grids for scattering amplitudes. It is demonstrated how the usage of deep-learning techniques enables accurate approximations for multidimensional amplitudes. A strategy for estimating how grid uncertainties propagate to physical observables is described. Finally, the development of the PYSECDEC and FTINT programs for the numerical evaluation of multi-loop integrals is presented.

Zusammenfassung

In den letzten Jahrzehnten, insbesondere seit der Entdeckung des Higgs-Bosons in 2012, ist die Notwendigkeit von präzisen Vorhersagen des Standardmodells für Experimente an Hochenergie-Teilchenbeschleunigern stetig gestiegen, da sich die experimentelle Genauigkeit, mit der Produktionsraten und Teilcheneigenschaften gemessen werden, erheblich verbessert hat. Um möglichst viele Informationen aus diesen Messungen zu gewinnen, muss die Präzision der theoretischen Vorhersagen auf einem vergleichbaren Niveau liegen. Eine dominante Quelle von Unsicherheiten sind fehlende Terme höherer Ordnung in der perturbativen Entwicklung von Streuamplituden. In dieser Dissertation werden zwei neuartige Techniken zur Berechnung von Mehrschleifen-Streuamplituden vorgestellt. Der Schwerpunkt liegt auf numerischen Methoden für Prozesse mit vielen kinematischen Skalen. Die erste Methode ist ein neues Verfahren zur Auflösung kinematischer Singularitäten von Feynmanintegralen durch deren Zerlegung in eine Summe von Integralen über explizit positive Integranden. Solcherart aufgelöste Integrale können im physikalischen Bereich ohne Konturdeformation ausgewertet werden, was die numerische Berechnung um bis zu mehrere Größenordnungen beschleunigt. Die zweite Methode beschäftigt sich mit dem Aufbau von Interpolationsgittern für Streuamplituden. Es wird gezeigt, wie der Einsatz von Deep-Learning-Techniken präzise Approximationen von mehrdimensionalen Amplituden ermöglicht. Eine Strategie zur Abschätzung der Fehlerfortpflanzung von Gitterunsicherheiten auf physikalische Observablen wird beschrieben. Abschließend wird die Entwicklung der Programme PYSECDEC und FTINT für die numerische Auswertung von Mehrschleifenintegralen vorgestellt.

Contents

1. Introduction	1
2. Theoretical Foundation	3
2.1. The Standard Model	3
2.1.1. Quantum chromodynamics	3
2.1.2. The electroweak theory	4
2.2. Perturbation theory	6
2.2.1. Factorisation	6
2.2.2. The hard process	6
2.2.3. Feynman diagrams and amplitudes	8
2.3. Renormalisation	11
2.3.1. QCD vacuum polarisation	12
2.3.2. The running coupling	14
2.3.3. Dimensional regularisation	16
2.4. Infrared divergences and cross sections	18
2.4.1. Cross sections	18
3. Feynman Integrals	23
3.1. The Feynman integral	24
3.1.1. Parametric representations	24
3.1.2. Divergences	28
3.2. Computing amplitudes	29
3.2.1. IBP reduction	31
3.2.2. Integral evaluation	33
3.3. Sector decomposition	37
3.3.1. Geometric sector decomposition	37
3.3.2. Numerical integration	38
3.3.3. pySecDec	40
3.4. Flow-time integrals with ftint	42
3.4.1. Calculation method	43
3.4.2. Using ftint	46
4. Positive Feynman Integrands	51
4.1. Preliminaries	52
4.2. Method	53
4.2.1. Overview	53
4.2.2. Algorithm for univariate bisectable integrals	55
4.2.3. Beyond univariate bisectable integrals	57
4.3. Examples	58
4.3.1. Massless examples	58
4.3.2. Massive examples	68
4.4. Benchmarks	80
4.4.1. 1-loop box with an off-shell leg	82
4.4.2. 1-loop on-shell pentagon	82
4.4.3. 2-loop non-planar boxes	83
4.4.4. 3-loop non-planar box	85

4.4.5.	1-loop triangle with an off-shell leg	86
4.4.6.	2-loop elliptic sunrise	88
4.4.7.	3-loop hyperelliptic banana	90
4.4.8.	Cancellations	92
4.5.	Conclusion and outlook	96
5.	Interpolating Amplitudes	99
5.1.	Setting the stage	100
5.1.1.	Defining the error	101
5.1.2.	The test functions	101
5.1.3.	Estimating the error	102
5.1.4.	Adaptivity and symmetries	103
5.2.	Polynomial interpolation	105
5.2.1.	Approximation error scaling	107
5.2.2.	Multiple dimensions	107
5.2.3.	Dimensionally adaptive grid	107
5.2.4.	Beyond the full grid	109
5.2.5.	Effects of noisy data	111
5.2.6.	Discussion	111
5.3.	B-spline interpolation	113
5.3.1.	B-spline basis functions	113
5.3.2.	B-spline interpolants	114
5.3.3.	Discussion	114
5.4.	Sparse grids	115
5.4.1.	Classical sparse grids	115
5.4.2.	Boundary treatment	118
5.4.3.	Spatially adaptive sparse grids	119
5.4.4.	Higher degree basis functions	120
5.4.5.	Discussion	122
5.5.	Machine learning techniques	125
5.5.1.	MLP	127
5.5.2.	L-GATr	127
5.5.3.	Discussion	130
5.6.	Conclusions	134
6.	Final Conclusion and Outlook	135
A.	Test functions	137
A.1.	Phase-space parametrisation for $pp \rightarrow t\bar{t}H$	137
A.2.	Amplitude definitions	138
	Acknowledgements	143
	References	145

CHAPTER 1

Introduction

At the Large Hadron Collider (LHC) at CERN in 2012, a massive scalar particle was discovered with properties that so far are consistent with the Standard Model (SM) prediction for the Higgs boson [1, 2]. Since the discovery, the ATLAS and CMS experiments have confirmed the existence of the Higgs boson in all of the main production modes and in five out of the seven main decay channels, $H \rightarrow b\bar{b}$, $\tau\bar{\tau}$, W^+W^- , ZZ , $\gamma\gamma$, while the remaining $H \rightarrow \mu^+\mu^-$ and $H \rightarrow Z\gamma$ channels have strong experimental signatures and are expected to be observed soon as well [3]. The experimental confirmation of the Higgs boson has cemented the SM as one of the greatest scientific achievements of the last century. It provides a unified description for three of the four fundamental forces; electromagnetism, the weak interaction and the strong force. The SM has successfully predicted the existence of several fundamental particles, such as the massive W^\pm and Z gauge bosons [4, 5] and the top quark [6, 7]. It has also passed a multitude of precision tests. For example, the theoretical prediction and experimental measurement for the electron magnetic moment agree within one part per trillion [8].

Despite its many great successes, the SM cannot be a complete description of nature. It does not incorporate gravity and thus fails to describe phenomena on cosmological scales. Combining the SM with general relativity into a unified field theory remains one of the main challenges of fundamental physics. Besides not explaining gravity, there are many other unsolved mysteries that the SM cannot account for, such as the nature of dark matter, the origin of the matter-antimatter asymmetry and the smallness of neutrino masses. There are countless Beyond SM (BSM) theories that address such discrepancies by extending the SM with new particles and interactions. But so far, there has been no direct evidence for any such model. A possible explanation for why this is the case, is that the energy scale of the new physics extension is beyond the reach of direct production at current colliders¹. In such a situation, the new particle must instead be probed through its indirect effects on SM processes. This is possible since

¹It is also possible that there exists light new physics that has escaped detection, such as axion-like particles [9].

it can still appear off shell in intermediate virtual states, and induce observable effects to SM processes in certain kinematic regions, such as in high transverse momentum regimes. By calculating precise predictions for event distributions in various observables, we obtain a method for inferring the existence of new heavy physics without having to produce it directly in the collider.

In order to disentangle a small excess of events from statistical fluctuations and background signals, very accurate theoretical predictions and experimental measurements are required. On the experimental side, tremendous progress on the accuracy of measurements has been achieved at the LHC. In particular in the Higgs sector, where Higgs couplings to third generation quarks and leptons, as well as to heavy gauge bosons, are currently known at the 5 – 10% level. In the near future, at the high-luminosity phase of the LHC, the available statistics will increase by a factor of 10 [3, 10], and the projection is for these bounds to be improved to the 2 – 4% level [11]. Currently, theoretical predictions are not able to match these precision levels [3]. There are many sources of theoretical uncertainty, for example related to fitting parton distribution functions, hadronization models, parton showers and missing higher-order perturbative corrections. The last mentioned is particularly relevant for processes with many kinematic scales, such as those involving multiple massive particles or with many particles in the final state.

One of the central ingredients in higher-order corrections is the scattering amplitude, which describes the transition probability between initial and final states. It is computed in the framework of perturbation theory, where the higher-order terms increase precision at the cost of being more complex to calculate. In order to bring the precision of theoretical predictions up to the standard of collider experiments, the computation of multi-loop virtual amplitudes is required. Such amplitudes involve the evaluation of multi-loop Feynman integrals. The main topic of this thesis is the development of computational methods for the evaluation of amplitudes and Feynman integrals at the multi-loop level, with a focus on processes that involve many kinematic scales.

This thesis is structured as follows. In Chapter 2 we provide an introduction to the necessary theoretical foundations required to understand precision calculations within the SM. The topics of factorisation, perturbation theory, loop corrections, regularisation and renormalisation are covered. In Chapter 3 a more detailed discussion on Feynman integrals is given, including an overview of standard computational methods for multi-loop amplitudes. The final sections of the chapter are dedicated to presenting the development of the programs PYSECDEC and FTINT, based on Ref. [12] and Ref. [13], respectively. In Chapter 4 a new method for resolving integrable singularities of Feynman integrals is presented, based on Ref. [14]. The new method makes it possible to avoid the standard procedure of contour deformation during numerical integration, which significantly boosts the performance of programs such as the previously covered PYSECDEC. Chapter 5 contains an investigation into the interpolation of multi-scale loop amplitudes, based on Ref. [15] and Ref. [16]. Several state-of-the-art approximation methods are studied in the context of amplitude interpolation. We also present a framework to estimate how grid errors propagate from amplitudes to observables. In Chapter 6 we conclude by summarizing the main results.

CHAPTER 2

Theoretical Foundation

This chapter introduces certain theoretical concepts that are required to understand precision calculations within the SM. The contents of this chapter are as follows; Sec. 2.1 defines the SM as a gauge theory to show the building blocks of scattering amplitudes. Sec. 2.2 introduces perturbation theory in terms of Feynman diagrams and describes how it can be applied to processes at hadron colliders, such as the LHC, through factorisation assumptions. Sec. 2.3 describes how to deal with infinities that appear in quantum field theory beyond leading order in perturbation theory. The framework of dimensional regularisation and the concept of ultraviolet renormalisation are discussed, as well as the running of coupling parameters. Finally, in Sec. 2.4, it is explained how to construct physical cross sections through cancellation of infrared divergences.

2.1. The Standard Model

The SM is a renormalisable quantum field theory (QFT) that is invariant under the gauge symmetry $SU(3)_C \times SU(2)_L \times U(1)_Y$. $SU(3)_C$ corresponds to Quantum Chromodynamics (QCD), which describes the strong interaction between quarks and gluons. The subscript indicates that the quark fields carry colour charges. $SU(2)_L \times U(1)_Y$ is the gauge group for the Electroweak (EW) theory. The subscript L denotes that the corresponding gauge transformations act only on left-handed fermions. The subscript Y is the hypercharge.

2.1.1. Quantum chromodynamics

QCD is a non-Abelian gauge theory described by the Lagrangian [17]

$$\begin{aligned} \mathcal{L}_{\text{QCD}} = & -\frac{1}{4}F_{\mu\nu}^a F^{a\mu\nu} + \sum_f \bar{q}_f(i\gamma_\mu D^\mu - m_f)q_f + \frac{1}{2\xi}\partial_\mu A^{a\mu}\partial^\nu A_\nu^a \\ & - \partial_\mu \bar{c}^a(\delta^{ac}\partial_\mu + g_s f^{abc}A_\mu^b)c^c, \end{aligned} \tag{2.1}$$

where the field-strength tensor is defined by

$$F_{\mu\nu}^a = \partial_\mu A_\nu^a - \partial_\nu A_\mu^a + g_s f^{abc} A_\mu^b A_\nu^c \quad (2.2)$$

and the covariant derivative is

$$D_\mu = \partial_\mu - ig_s A_\mu^a t^a, \quad (2.3)$$

where the t^a are the generators of $SU(3)$ in the fundamental representation. The first term in the Lagrangian is the kinetic term for the gluons. It involves a factor $g_s f^{abc} A_\mu^b A_\nu^c$ which encodes a characteristic feature of non-Abelian theories, namely the presence of self-interactions among the gauge bosons. The second term is a sum over quark flavours, where m_f is the mass of the quark of flavour f . It includes the covariant derivative, which generates the interactions between gluons and quarks through the $g_s A_\mu^a t^a$ term.

The latter two terms are technical insertions to remove unphysical degrees of freedom from the theory. The third term is a gauge-fixing term and ξ is a so-called gauge parameter. Its value is arbitrary and must not affect physical predictions. A common choice is the Feynman gauge where $\xi = 1$, since this tends to simplify calculations. The fourth term involves the Faddeev-Popov ghost fields [18], which is a gauge dependent term that is necessary to ensure unitarity. Ghost fields are unphysical and only appear as virtual states. They are constructed such that they exactly cancel the unphysical degrees of freedom corresponding to longitudinal and time-like polarisations of gluons. Additionally, the ghost fields make the QCD Lagrangian invariant under the BRST symmetry, which ensures that QCD is renormalisable [19, 20]. The inclusion of ghost fields can be avoided by working in axial (physical) gauges, though this is usually not done as it increases computational complexity.

The QCD Lagrangian can be used to derive Feynman rules, which are the building blocks of scattering amplitudes in perturbation theory. See Table 2.1 in Sec. 2.2.3 for a selection of QCD Feynman rules, with the convention used in Ref. [17].

2.1.2. The electroweak theory

The EW theory is, like QCD, a non-Abelian gauge theory which is described by the Lagrangian [21]

$$\mathcal{L}_{\text{EW}} = -\frac{1}{4} W_{\mu\nu}^a W^{a\mu\nu} - \frac{1}{4} B_{\mu\nu} B^{\mu\nu} + (D_\mu H)^\dagger (D_\mu H) + m^2 H^\dagger H - \lambda (H^\dagger H)^2, \quad (2.4)$$

where the covariant derivative is defined by

$$D_\mu H = \partial_\mu H - ig W_\mu^a \tau^a H - \frac{1}{2} ig' B_\mu H. \quad (2.5)$$

The W_μ^a are the three gauge bosons of $SU(2)$, B_μ is the hypercharge gauge boson and H is the Higgs doublet. The Higgs potential can be read off from the Lagrangian

$$V(H) = -m^2 |H|^2 + \lambda |H|^4, \quad m^2, \lambda > 0. \quad (2.6)$$

It has a minimum at $|H|^2 = \frac{m^2}{2\lambda} = v^2$, where v is the vacuum expectation value. In the Standard Model, the $SU(2)_L \times U(1)_Y$ symmetry is spontaneously broken through

the Brout–Englert–Higgs mechanism [22, 23]. The symmetry breaking follows from the invariance of the non-zero vacuum expectation value under the $SU(2)$ transformations. In the unitary gauge, the Higgs doublet can be parameterised around its vacuum expectation value according to [17, 21]

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ v + h \end{pmatrix}, \quad (2.7)$$

where h is the Higgs boson. The gauge-boson mass terms are found by taking the square of Eq. (2.5), substituting the parameterised H , and considering the terms proportional to v

$$|D_\mu H|^2 = g^2 \frac{v^2}{8} \left[(W_\mu^1)^2 + (W_\mu^2)^2 + \left(\frac{g'}{g} B_\mu - W_\mu^3 \right)^2 \right]. \quad (2.8)$$

The result is three massive gauge bosons

$$W_\mu^\pm = \frac{1}{\sqrt{2}} (W_\mu^1 \mp iW_\mu^2)$$

$$Z_\mu^0 = \frac{1}{\sqrt{g^2 + g'^2}} (gW_\mu^3 - g'B_\mu),$$

as well as a massless gauge boson

$$A_\mu^0 = \frac{1}{\sqrt{g^2 + g'^2}} (g'W_\mu^3 + gB_\mu).$$

These correspond exactly to the observed W^\pm and Z bosons, and to the photon of electromagnetism. The masses of the W^\pm and Z bosons are related to the vacuum expectation value, and to the couplings, according to

$$m_W = g \frac{v}{2}, \quad m_Z = \sqrt{g^2 + g'^2} \frac{v}{2}. \quad (2.9)$$

They are not independent, and their relation is given by

$$m_W = m_Z \cos \theta_w, \quad \tan \theta_w = \frac{g'}{g}, \quad (2.10)$$

where θ_w is the weak mixing angle. In this way, θ_w and m_W , together with the electric charge e , are enough to specify all tree-level processes involving an exchange of a W or Z boson [17]. In this sense, the set (e, θ_w, m_W) comprises an electroweak input scheme. This means that by measuring these three parameters experimentally, the values of all remaining electroweak parameters are predicted by the theory. The relation between the electric charge and the g and g' couplings is given by

$$e = g \sin \theta_w = g' \cos \theta_w. \quad (2.11)$$

The electroweak Lagrangian can be used to derive Feynman rules for the electroweak theory. Here, we are mainly interested in the Yukawa couplings between the Higgs and heavy quarks. The Higgs doublet provides a way of including quark mass terms that do not spoil $SU(2)$ gauge invariance

$$\mathcal{L}_{\text{mass}} = -\lambda_d^d \bar{Q}^i H d_R^j - Y_{ij}^u \bar{Q}^i \tilde{H} u_R^j + h.c., \quad \tilde{H} = i\sigma_2 H^*. \quad (2.12)$$

Since the examples and applications in this thesis are limited to QCD processes, we refer to standard texts [17, 21] for a full list of SM Feynman rules.

2.2. Perturbation theory

To calculate predictions for processes at high-energy colliders the most commonly used framework is that of perturbation theory. This section introduces the concepts of factorisation, scattering amplitudes and their perturbative expansion in terms of Feynman diagrams.

2.2.1. Factorisation

At hadron-hadron colliders the particle beams consist of composite particles, such as protons at the LHC. Such particles have high masses and provide the collider with a high energy reach, which enables direct production of heavy elementary particles. This comes at the price of a more complicated theoretical description. The reason is that the hadrons are QCD bound states that involve long-range interactions, while the hard interaction between the colliding partons is short-range. Predictions for all processes at hadron colliders therefore rely on the concept of *factorisation*. The basic idea is that physics at very different energy scales should decouple. In hadron-hadron collisions, cross sections σ are computed through the collinear factorisation formula² [24]

$$\sigma = \sum_{a,b} \int_0^1 dx_a dx_b f_{a/h_1}(x_a) f_{b/h_2}(x_b) d\hat{\sigma}_{ab \rightarrow n}, \quad (2.13)$$

where $\hat{\sigma}_{ab \rightarrow n}$ is the partonic cross section that describes the hard interaction between partons a and b and the $f_{a/h}(x_a)$ are Parton Distribution Functions (PDFs) that encode the long-range interactions in the hadron. At leading order $f_{a/h}(x_a)$ describes the probability of finding parton a in hadron h with a fraction x_a of the total hadron momentum. The low energy scale of the long-range interactions in the hadron means the PDFs are inherently non-perturbative objects. They can thus not be computed in the framework of perturbation theory and must instead be fitted from experimental data. Assuming factorisation holds, PDFs are process independent. They can thus be fitted with data from precisely known processes, that are easy to compute and measure, and then be applied to other processes. The evolution of PDFs between different energy scales can be calculated perturbatively using the Dokshitzer–Gribov–Lipatov–Altarelli–Parisi (DGLAP) equations [25–27].

That QCD interactions can be decomposed into a convolution of long- and short-distance components is a highly non-trivial result. In fact, Eq. (2.13) has only been rigorously proven for specific processes such as deep inelastic scattering [25] and Drell-Yan [28, 29]. Moreover, it is only an exact relation up to so-called power corrections $\mathcal{O}((\Lambda/Q)^p)$, where Q is a characteristic energy scale of the process. In the limit $Q^2 \rightarrow \infty$ the power corrections vanish, which makes it a good description for high-energy collisions.

2.2.2. The hard process

A high-energy collision between elementary particles, such as the partons coming out of a hadron, is known as a hard interaction, and is described by a partonic cross section according to

$$\hat{\sigma}_{ab \rightarrow n} = \frac{1}{2\hat{s}} \int d\Phi_n |\mathcal{M}_{ab \rightarrow n}(p_1, \dots, p_n)|^2, \quad (2.14)$$

²This equation describes specifically $2 \rightarrow n$ scattering, which is the general case at collider experiments.

where $d\Phi_n$ is the n -particle Lorentz-Invariant-Phase-Space (LIPS) defined as

$$d\Phi_n = (2\pi)^4 \delta^4(q_a + q_b - \sum_i p_i) \prod_{i=1}^n \frac{d^4 p_i}{(2\pi)^4} 2\pi \delta(p_i^2 - m_i^2) \theta(p_i^{(0)}). \quad (2.15)$$

The $\delta^4(q_a + q_b - \sum_i p_i)$ imposes momentum conservation between the initial and final states, $\delta(p_i^2 - m_i^2)$ is an on-shell condition for the final-state particles and $\theta(p_i^{(0)})$ ensures that the final-state particles have positive energy. The prefactor in Eq. (2.14) is known as the flux factor and is related to the centre-of-mass energy s of the underlying hadron collision by $\hat{s} = x_a x_b s$. The $\mathcal{M}_{ab \rightarrow n}(p_1, \dots, p_n)$ is the main object of interest in this thesis and is known as the Feynman amplitude. It is the non-trivial part of the S -matrix [30] that describes the transition probability between an initial state i and a final state f ³

$$\langle f | S - \mathbb{1} | i \rangle = i (2\pi)^4 \delta^4(q_a + q_b - \sum_i p_i) \mathcal{M}_{i \rightarrow f}(p_1, \dots, p_n). \quad (2.16)$$

It is a complex-valued function and its square can be interpreted as a probability density that, when integrated over a phase-space region, describes the probability of producing the final state n in that region. The amplitude depends on a coupling parameter g , which essentially is the interaction strength between particles. In a weakly interacting theory with small g , the amplitude can be expanded perturbatively as

$$\mathcal{M} = \sum_{k=0}^{\infty} g^{2k} \mathcal{M}_k, \quad (2.17)$$

where the first term \mathcal{M}_0 may or may not already include powers of g . Exactly what the \mathcal{M}_k are is best understood in the context of Feynman diagrams which is described in Sec. 2.2.3. The higher order terms are suppressed by increasing powers of the coupling⁴. Naturally, the series converges more quickly the smaller the coupling is. The first non-zero term in this expansion is referred to as the Leading-Order (LO) amplitude, the second the Next-to-Leading-Order (NLO) contribution to the amplitude, and so on. The expansion of \mathcal{M} suggests that the cross section can also be decomposed order-by-order as

$$\hat{\sigma} = \hat{\sigma}_{\text{LO}} + \alpha \hat{\sigma}_{\text{NLO}} + \alpha^2 \hat{\sigma}_{\text{NNLO}} + \dots, \quad (2.18)$$

where $\alpha = g^2/4\pi$. Sec. 2.4 describes in detail what amplitude ingredients must enter the cross section at the different orders. For now we state that the higher-order terms, that can be interpreted as quantum corrections to the Born-level scattering process, increase precision at the cost of being more complex to calculate. In practice the sum has to be truncated at a finite order. This gives rise to dependence on the unphysical scales μ_R and μ_F for both the amplitude and the cross section, such that

$$\begin{aligned} \mathcal{M}(p_1, \dots, p_n) &\rightarrow \mathcal{M}(p_1, \dots, p_n; \mu_F, \mu_R), \\ \hat{\sigma} &\rightarrow \hat{\sigma}(\mu_F, \mu_R). \end{aligned}$$

³We always use the shorthand $\langle f | \mathcal{M} | i \rangle = \mathcal{M}_{i \rightarrow f}$ and often also $\mathcal{M}_{i \rightarrow f} = \mathcal{M}$ unless it is not clear from the context which process we are referring to.

⁴The number of terms entering into \mathcal{M}_k grows factorially which means the suppression from g^{2k} is eventually overtaken and the expansion starts to diverge [31]. This only happens at very high orders in the theories considered in this thesis, and is therefore of no practical relevance here.

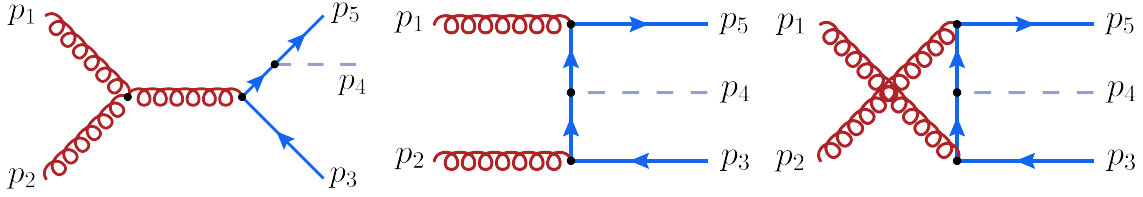


Figure 2.1.: Example tree-level diagrams for $gg \rightarrow t\bar{t}H$. The red curly lines are gluons, the blue solid lines are top quarks and the dashed line is the Higgs boson. All other tree-level diagrams differ from these three only by which top-quark line radiates the Higgs boson [34]. External momenta are pointing inward.

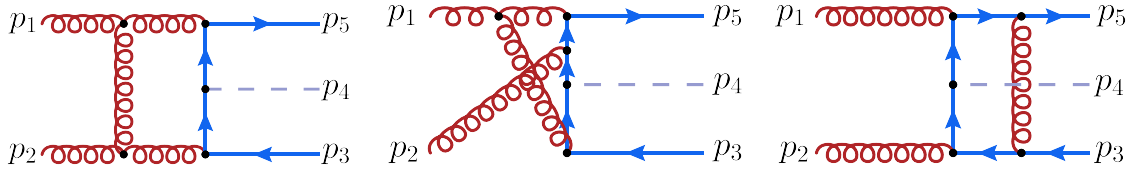


Figure 2.2.: Example 1-loop diagrams for $gg \rightarrow t\bar{t}H$. The red curly lines are gluons, the blue solid lines are top quarks and the dashed line is the Higgs boson. External momenta are pointing inward.

The subscripts refer to renormalisation and factorisation which are discussed in Sec. 2.3 and Sec. 2.4, respectively. The presence of μ_R and μ_F implies that there is a scale uncertainty on the cross section, associated with the truncation of the perturbative expansion. Reducing this uncertainty is the underlying motivation for the development of the computational methods presented in this thesis.

2.2.3. Feynman diagrams and amplitudes

The \mathcal{M}_k in Eq. (2.17) can be represented pictorially by Feynman diagrams and translated to amplitude expressions through Feynman rules. To demonstrate how this works we consider, as an example, the QCD process of $gg \rightarrow t\bar{t}H$ up to NLO. The LO amplitude is represented by Feynman diagrams of the type shown in Fig. 2.1⁵ [34]. There are five additional diagrams that only differ in the location of the Higgs boson radiated off the top. These are *tree-level* diagrams, which means they do not include any closed loops of virtual particles. In Fig. 2.2 a selection of diagrams contributing to the amplitude at NLO is shown. These are *loop* diagrams since in each case they include a closed loop of virtual particles. To each such loop, there is an associated *loop integral* that must be evaluated to obtain the amplitude. The loop integrals usually constitute the main obstacle in obtaining higher-order amplitudes, and they are discussed in detail in Chapter 3. For now, we can finally define the terms \mathcal{M}_k as the sum of all diagrams containing k loops (or the radiation of up to k extra particles, see Sec. 2.4). For some processes there are no tree-level diagrams that contribute to the amplitude, which means that the LO consists of 1-loop diagrams. Such processes are said to be loop induced.

Feynman diagrams are translated to amplitude expressions through Feynman rules. They are of two main types: vertex rules and propagator rules. In the diagrams of

⁵All diagrams in this thesis have been drawn using FEYNGAME [32, 33].

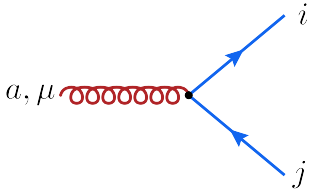
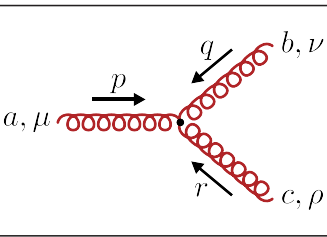
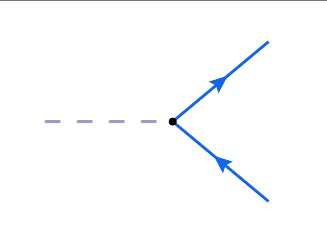
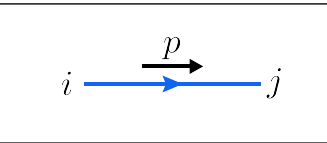
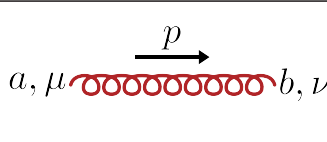
	$ig_s \gamma_\mu t_{ij}^a$
	$g_s f^{abc} \begin{bmatrix} +g^{\mu\nu}(p-q)^\rho \\ +g^{\nu\rho}(q-r)^\mu \\ +g^{\rho\mu}(r-p)^\nu \end{bmatrix}$
	$-i \frac{g}{2} \frac{m_f}{m_W}$
	$\frac{i(\not{p} + m_f)}{p^2 - m_f^2 + i\delta}$
	$\frac{-i\delta_{ab}}{p^2 + i\delta} \left(g^{\mu\nu} - (1 - \xi) \frac{p^\mu p^\nu}{p^2} \right)$

Table 2.1.: Selection of QCD Feynman rules [17]. The red curly lines are gluons, the blue solid lines are (top) quarks and the dashed line is the Higgs boson. The gluon propagator is given in covariant gauge. The choice $\xi = 1$, for example, corresponds to the Feynman gauge.

Fig. 2.1 and Fig. 2.2 the relevant vertices are respectively: the Fermion-gauge interaction between a gluon and two top quarks, the triple-gauge interaction between three gluons and the Yukawa interaction between the Higgs boson and the top quarks. The relevant propagator rules are those of the top-quark and gluon propagators respectively. The vertex and propagator rules are summarized in Table 2.1 [17]. The g_s and g_t are the strong and weak couplings respectively, $g^{\mu\nu}$ is the Minkowski metric, the γ^μ are Dirac gamma matrices and $\not{p} = p_\mu \gamma^\mu$. The convention for momentum directions is all incoming, such that momentum conservation implies $\sum_i p_i = 0$. Moreover, to each initial-state (anti)fermion and gauge boson there are associated spinors $u^s(p)$ ($\bar{v}^s(p)$) and polarisation vectors $\epsilon_\mu(p)$ respectively. Final-state particles have conjugated spinors and polarisation vectors. Finally, diagrams including N closed fermion loops should be multiplied by a factor $(-1)^N$. A full list of the SM Feynman rules can be found in standard texts such as Ref. [17].

To see how the Feynman rules can be applied in practice we return to the leading-order $gg \rightarrow t\bar{t}H$ amplitude, which consists of eight tree-level diagrams, three of which can be seen in Fig. 2.1. The five remaining diagrams only differ in which top-quark line

radiates the Higgs boson [34, 35] and it is natural to organise the contributions in terms of s -, t - and u -channel type diagrams

$$\mathcal{M}_{0,s}^{\mu\nu} = \mathcal{M}_{0,s}^{(1),\mu\nu} + \mathcal{M}_{0,s}^{(2),\mu\nu} \quad (2.19)$$

$$\mathcal{M}_{0,t}^{\mu\nu} = \mathcal{M}_{0,t}^{(1),\mu\nu} + \mathcal{M}_{0,t}^{(2),\mu\nu} + \mathcal{M}_{0,t}^{(3),\mu\nu} \quad (2.20)$$

$$\mathcal{M}_{0,u}^{\mu\nu} = \mathcal{M}_{0,u}^{(1),\mu\nu} + \mathcal{M}_{0,u}^{(2),\mu\nu} + \mathcal{M}_{0,u}^{(3),\mu\nu}, \quad (2.21)$$

where in each case the polarisation vectors and spinors are factored such that $\mathcal{M}_{0,i} = \epsilon_\mu(p_1) \epsilon_\nu(p_2) \bar{u}(p_5) v(p_3) \mathcal{M}_{0,i}^{\mu\nu}$, $i = s, t, u$. Using the Feynman rules from Table 2.1, the three diagrams from Fig. 2.1 can be translated into amplitude expressions

$$i\mathcal{M}_{0,s}^{(1),\mu\nu} = -g_s^2 g_t f^{abc} t^c \frac{\not{p}_4 + \not{p}_5 + m_t}{p_{45}^2 - m_t^2 + i\delta} \gamma_\delta \frac{1}{p_{12}^2 + i\delta} \quad (2.22)$$

$$\times [g^{\mu\nu}(p_1 - p_2)^\delta + g^{\nu\delta}(p_1 + 2p_2)^\mu - g^{\delta\mu}(2p_1 + p_2)^\nu], \quad (2.23)$$

$$i\mathcal{M}_{0,t}^{(1),\mu\nu} = ig_s^2 g_t t^a t^b \gamma^\mu \frac{\not{p}_1 + \not{p}_5 + m_t}{p_{15}^2 - m_t^2 + i\delta} \frac{\not{p}_2 + \not{p}_3 + m_t}{p_{23}^2 - m_t^2 + i\delta} \gamma^\nu, \quad (2.24)$$

$$i\mathcal{M}_{0,u}^{(1),\mu\nu} = ig_s^2 g_t t^b t^a \gamma^\mu \frac{\not{p}_2 + \not{p}_5 + m_t}{p_{25}^2 - m_t^2 + i\delta} \frac{\not{p}_1 + \not{p}_3 + m_t}{p_{13}^2 - m_t^2 + i\delta} \gamma^\nu, \quad (2.25)$$

where $p_{ij} = p_i + p_j$, $\xi = 1$ (Feynman gauge), a, b are colour indices and the spinor indices are left implicit. We also use $g_t = \frac{g m_t}{2m_W}$ to denote the coupling between the top quark and the Higgs boson. The remaining five diagrams yield similar expressions, just with different momenta in the top-quark propagators. The leading-order amplitude for $gg \rightarrow t\bar{t}H$ is thus compactly expressed as

$$\mathcal{M}_0^{\mu\nu} = \sum_{i=1}^2 \mathcal{M}_{0,s}^{(i),\mu\nu} + \sum_{i=1}^3 \mathcal{M}_{0,t}^{(i),\mu\nu} + \sum_{i=1}^3 \mathcal{M}_{0,u}^{(i),\mu\nu}. \quad (2.26)$$

The next step is to square the amplitude, which involves polarisation and spin sums. In this case, it is necessary to average over the initial-state gluon polarisations, since they cannot be measured at experiments. If we are calculating unpolarised cross sections we should also sum over final-state spins. Moreover, since colour is not measured we must average and sum over colours in the initial and final state respectively. For this example, the object we are actually interested in is the polarisation-colour-averaged, spin-colour-summed squared amplitude. We make the replacement

$$|\mathcal{M}_0^{\mu\nu}|^2 \rightarrow \frac{1}{256} \sum_{\text{cols}} \sum_{s_1, s_2} \sum_{\lambda_1, \lambda_2} |\mathcal{M}_0^{\mu\nu}|^2, \quad (2.27)$$

where s_1, s_2 are the spins of the final-state quarks, λ_1, λ_2 are the polarisations of the initial-state gluons and the factor in front is $N_{\text{spins}}^2 N_{\text{cols}}^2 = 2^2 \cdot 8^2$. The spin sums follow from the completeness relations. In this case they take the form

$$\sum_{s_1} u^{s_1}(p_5) \bar{u}^{s_1}(p_5) = \not{p}_5 + m_t, \quad (2.28)$$

$$\sum_{s_2} v^{s_2}(p_3) \bar{v}^{s_2}(p_3) = \not{p}_3 - m_t, \quad (2.29)$$

for the top and antitop quarks respectively. The polarisation sum for the gluon with momentum p_1 is

$$\sum_{\lambda} \epsilon_{\lambda}^{\mu}(p_1) \epsilon_{\lambda}^{\star \nu}(p_1) = -g^{\mu\nu} + \frac{p_1^{\mu} n^{\nu} + p_1^{\nu} n^{\mu}}{p_1 \cdot n}, \quad (2.30)$$

where n^{μ} is a light-like vector, dual to p_1^{μ} . For QED-like processes, or if ghost contributions are included, the second term can be dropped and the polarisation sums result in metric tensors, that contract the indices in $\mathcal{M}_0^{\mu_1 \mu_2} (\mathcal{M}_0^{\nu_1 \nu_2})^{\dagger}$ to leave a scalar $|\mathcal{M}_0|^2$. Calculating \mathcal{M}_0^2 generically requires interfering all diagrams with each other. A more convenient approach for this particular case is used in Ref. [35], where contributions are organised in terms of symmetric and antisymmetric colour factors. Defining $\mathcal{A}_{0,i}$ as the coefficient of the colour structure in $\mathcal{M}_{0,i}$, the amplitude is expressed as

$$\mathcal{M}_0 = \mathcal{A}_0^{\text{sym}} (t^a t^b + t^b t^a) + \mathcal{A}_0^{\text{asym}} (t^a t^b - t^b t^a), \quad (2.31)$$

with

$$\mathcal{A}_0^{\text{sym}} = \frac{1}{2} (\mathcal{A}_{0,t} + \mathcal{A}_{0,u}), \quad \mathcal{A}_0^{\text{asym}} = \mathcal{A}_{0,s} + \frac{1}{2} (\mathcal{A}_{0,t} - \mathcal{A}_{0,u}), \quad (2.32)$$

where the commutation relation $[t^a, t^b] = i f^{abc} t^c$ for the SU(3) generators is used to eliminate the structure constants from the s -channel amplitude. Using that the symmetric and antisymmetric colour factors are orthogonal, a compact representation of the squared leading-order amplitude is obtained as [35]

$$|\mathcal{M}_0|^2 = \frac{N_c(N_c^2 - 1)}{2} (|\mathcal{A}_0^{\text{asym}}|^2 + |\mathcal{A}_0^{\text{sym}}|^2) + \frac{N_c^2 - 1}{N_c} |\mathcal{A}_0^{\text{sym}}|^2, \quad (2.33)$$

where N_c is the number of colours. The polarisation-colour-averaged, spin-colour-summed squared amplitude can now be integrated over the phase space and convoluted with parton distribution functions, to yield the leading-order cross section σ_{LO} . To obtain higher-order cross sections, 1-loop diagrams of the type in Fig. 2.2 have to be computed, as well as real radiation diagrams, yielding corrections suppressed by α_s relative to the LO. These diagrams include an unconstrained momenta that must be integrated over. In general, these integrals yield infinities and are not immediately well defined. The next section will describe how divergences appearing in loop integrals are treated in quantum field theory calculations.

2.3. Renormalisation

At higher orders in perturbation theory it is necessary to calculate loop integrals to obtain scattering amplitudes. In general, these integrals are by default not well defined and can for example diverge in the ultraviolet (UV) regime, corresponding to large loop momentum⁶. Such divergences are treated through regularisation and renormalisation procedures. To introduce these concepts, we consider first a 1-loop contribution to the QCD vacuum polarization.

⁶Loop integrals can also have singularities in the infrared regime, corresponding to small loop momentum. Such divergences are discussed in Sec. 2.4.

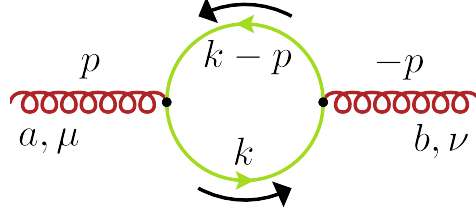


Figure 2.3.: Light-quark loop contribution to the vacuum polarisation of the gluon. External momenta are incoming.

2.3.1. QCD vacuum polarisation

At one loop, there are four diagrams that contribute to the correction of the gluon propagator. To first demonstrate how UV divergences appear we consider in detail the light-quark-loop contribution, see Fig. 2.3. The remaining three vacuum polarisation diagrams can be seen in Fig. 2.5.

Applying the Feynman rules from Table 2.1 gives the expression for the amplitude as

$$i\mathcal{M}_{ab}^{\mu\nu} = -(ig_s)^2 \text{Tr} \{t^a t^b\} \int \frac{d^4 k}{(2\pi)^4} \frac{\text{Tr} \{ \gamma_\nu i \not{k} \gamma_\mu i (\not{k} - \not{p}) \}}{k^2 (k-p)^2}, \quad (2.34)$$

where p is the gluon momentum and the Feynman $i\delta$ prescription has been left implicit. The polarisation vectors are left out such that the loop correction can be incorporated directly into the propagator. The colour trace is

$$\text{Tr} \{t^a t^b\} = \frac{1}{2} \delta_{ab}. \quad (2.35)$$

There are two traces of gamma matrices that have to be evaluated

$$\text{Tr} \{ \gamma_\nu \not{k} \gamma_\mu \not{k} \} = k^\rho k^\delta \text{Tr} \{ \gamma_\nu \gamma_\rho \gamma_\mu \gamma_\delta \} = 4k^\rho k^\delta (g^{\nu\rho} g^{\mu\delta} - g^{\mu\nu} g^{\rho\delta} + g^{\mu\rho} g^{\nu\delta}) \quad (2.36)$$

$$= 4(2k^\mu k^\nu - k^2 g^{\mu\nu}), \quad (2.37)$$

$$\text{Tr} \{ \gamma_\nu \not{k} \gamma_\mu \not{p} \} = k^\rho p^\delta \text{Tr} \{ \gamma_\nu \gamma_\rho \gamma_\mu \gamma_\delta \} = 4k^\rho p^\delta (g^{\nu\rho} g^{\mu\delta} - g^{\mu\nu} g^{\rho\delta} + g^{\mu\rho} g^{\nu\delta}) \quad (2.38)$$

$$= 4(k^\mu p^\nu + k^\nu p^\mu - g^{\mu\nu} k \cdot p). \quad (2.39)$$

Inserting the traces into the amplitude gives

$$i\mathcal{M}_{ab}^{\mu\nu} = -\delta_{ab} g_s^2 \int \frac{d^4 k}{8\pi^4} \frac{2k^\mu k^\nu - k^2 g^{\mu\nu} - k^\mu p^\nu - k^\nu p^\mu + g^{\mu\nu} k \cdot p}{k^2 (k-p)^2}. \quad (2.40)$$

This integral is divergent, which can be seen explicitly by studying the terms individually. For example, the second term contains the following contribution

$$\int \frac{d^4 k}{(k-p)^2} = -i \int \frac{d^4 k_E}{k_E^2} = -2i\pi^2 \int_0^\infty dr r, \quad (2.41)$$

where a shift $k \rightarrow k+p$ has been performed, followed by a Wick rotation to Euclidean space and an integration over the angular coordinates. This integral is clearly divergent in the $r \rightarrow \infty$ limit. Evaluating it requires introducing a regulator, which can be done in many different ways. For example taking Λ as a UV cutoff results in

$$-2i\pi^2 \int_0^\infty dr r \rightarrow -2i\pi^2 \int_0^\Lambda dr r = -i\pi^2 \Lambda^2. \quad (2.42)$$

The original integral is restored by taking the limit $\Lambda \rightarrow \infty$. The divergence in Eq. (2.41) has been regulated in the sense that the singularity has been made explicit, it diverges quadratically. Gauge theories, such as QCD, are known to diverge at most logarithmically. Indeed, evaluating the full integral, as well as the other diagrams in Fig. 2.5, results in a regulator-dependent expression that diverges only logarithmically upon taking $\Lambda \rightarrow \infty$ [21].

This divergence is not as serious of a problem as it may first seem, as $\mathcal{M}_{\mu\nu}^{ab}$ itself is not measurable. The computation is well defined as long as the dependence on Λ drops out when $\mathcal{M}_{\mu\nu}^{ab}$ is used in the computation of a physical observable. To cancel UV divergences, such as the ones found in the QCD vacuum polarisation computation, the key is to realise that there are already multiple unphysical parameters in the theory. These are the bare couplings, fields and masses as they appear in the Lagrangian, and they may be redefined to absorb a finite number of UV divergences. The redefinition of field parameters to absorb UV divergences is known as *renormalisation*.

To make this explicit, we consider QCD, where there are four parameter redefinitions corresponding to fields and masses [21]

$$A_R^\mu = Z_3^{-\frac{1}{2}} A_0^\mu, \quad c_R = Z_{3,c}^{-\frac{1}{2}} c_0, \quad \psi_R = Z_2^{-\frac{1}{2}} \psi_0, \quad m_R = Z_m^{-1} m. \quad (2.43)$$

Since the strong coupling g_s appears in four different interaction vertices in \mathcal{L}_{QCD} , there are four possible equations for its redefinition

$$\begin{aligned} g_R &= g_0^{q\bar{q}g} Z_1^{-1} Z_2 Z_3^{1/2}, & g_R &= g_0^{c\bar{c}g} Z_{1,c}^{-1} Z_{3,c} Z_3^{1/2}, \\ g_R &= g_0^{3g} Z_{A^3}^{-1} Z_3^{3/2}, & g_R &= g_0^{4g} Z_{A^4}^{-1/2} Z_3. \end{aligned} \quad (2.44)$$

The Z_i are known as renormalisation constants. Since there are only five bare parameters but eight equations, there must be relations between the renormalisation constants. These are known as the Slavnov-Taylor identities and are given by [36, 37]

$$\frac{Z_{A_3}}{Z_3} = \frac{Z_{1c}}{Z_{3c}} = \frac{Z_1}{Z_2} = \frac{Z_{A^4}}{Z_{A_3}}. \quad (2.45)$$

The renormalisation constants are usually parameterised by counterterms δ_i . For QCD, there are eight counterterms, defined by [17]

$$\begin{aligned} \delta_2 &= Z_2 - 1, & \delta_3 &= Z_3 - 1, & \delta_{3c} &= Z_{3c} - 1, & \delta_m &= Z_m m_0 - m_R, \\ \delta_1 &= \frac{g_0}{g_R} Z_2 Z_3^{1/2} - 1, & \delta_{A^3} &= \frac{g_0}{g_R} Z_3^{3/2} - 1, & \delta_{A^4} &= \frac{g_0^2}{g_R^2} Z_3^2 - 1, & \delta_{1c} &= \frac{g_0}{g_R} Z_{3c} Z_3^{1/2} - 1. \end{aligned}$$

In practice, theories are renormalised systematically in the framework of renormalised perturbation theory. It is formulated by inserting the redefinitions from Eq. (2.43) and Eq. (2.44) into the Lagrangian. Expanding the renormalisation constants in counterterms results in additional terms in the Lagrangian that can be used to derive Feynman rules and diagrams. The general structure of the Lagrangian becomes

$$\mathcal{L} = \mathcal{L}_{\text{ren}} + \mathcal{L}_{\text{c.t.}}, \quad (2.46)$$

where \mathcal{L}_{ren} is the original Lagrangian with bare parameters replaced by renormalised ones. The $\mathcal{L}_{\text{c.t.}}$ contains the counterterms. For QCD, it takes the form [17]

$$\begin{aligned}\mathcal{L}_{\text{c.t.}} = & -\frac{1}{4}\delta_3(\partial_\mu A_\nu^a - \partial_\nu A_\mu^a)^2 + \bar{q}(i\delta_2\not{\partial} - \delta_m)q - \delta_{3,c}\bar{c}^a\partial_\mu\partial^\mu c^a \\ & + g_0\delta_1 A_\mu^a \bar{q}\gamma^\mu q - g_0\delta_{A^3} f^{abc}(\partial_\mu A_\nu^a)A_\mu^b A_\nu^c \\ & - g_0^2\delta_{A^4}(f^{eab}A_\mu^a A_\nu^b)(f^{ecd}A_\mu^c A_\nu^d) - g_0\delta_{1,c}\bar{c}^a f^{abc}\partial^\mu A_\mu^b c^c,\end{aligned}\tag{2.47}$$

where g_0 is the bare strong coupling. In this way, the values of the counterterms can be chosen such that the additional diagrams from $\mathcal{L}_{\text{c.t.}}$ exactly cancel the UV divergences of the loop integrals coming from \mathcal{L}_{ren} .

There is also freedom in modifying the finite part of the result, which corresponds to different *renormalisation schemes*. The most basic scheme where only the divergences are removed is called the minimal subtraction (MS) scheme. A more common choice is the modified minimal subtraction ($\overline{\text{MS}}$) scheme, where in addition to the divergences $\ln(4\pi) - \gamma_E$ is subtracted from the finite part. This is a convenient choice because these terms always appear in loop integrals when working in dimensional regularisation. Another common choice is the on-shell scheme, where the finite parts of the counterterms are chosen such that the renormalised mass is equal to the pole mass. The pole mass is the location of the pole in the renormalised propagator and can be interpreted as the physical mass of the particle. For fixed-order computations, there can be a renormalisation scheme uncertainty associated to the result, for example due to different choices for the quark-mass renormalisation.

2.3.2. The running coupling

For the perturbative expansion in Eq. (2.17) to be well defined and converge quickly, a necessary⁷ requirement is that the coupling α must be small enough. The whole machinery with Feynman rules and diagrams is built on the assumption that including at most a few \mathcal{M}_k , yields a sufficiently accurate approximation of \mathcal{M} for phenomenological applications. Stating whether a coupling parameter is small enough for a theory to be well described perturbatively is not straightforward due to loop corrections. It is explained in Sec. 2.3 how loop corrections produce UV divergences that necessitate a regularisation and renormalisation procedure. The result is that a dependence on an unphysical renormalisation scale is induced to the coupling such that $\alpha := \alpha(\mu_R)$ [38]. The scale dependence of $\alpha(\mu_R)$ is referred to as the *running* of the coupling. It is thus implied that the validity and rate of convergence of the perturbative expansion may depend on the energy scale of the interaction.

The running of the coupling is more severe in some theories than others. For example, in QED, the fine structure constant changes only by about 10% in the range from macroscopic distances down to the shortest distances of high-energy colliders [38]. Since its value at low energies is approximately 1/137, perturbation theory for QED can be used effectively for all experimentally reachable energy scales⁸. In QCD, on the other

⁷But not sufficient. Additionally, the higher-order amplitudes \mathcal{M}_k must also be small enough to not spoil convergence. For example, if each power in α is accompanied by large logarithms, this necessitates all-order resummation.

⁸Far beyond the energy reach of collider experiments, perturbation theory for QED breaks down due to the presence of a Landau pole.

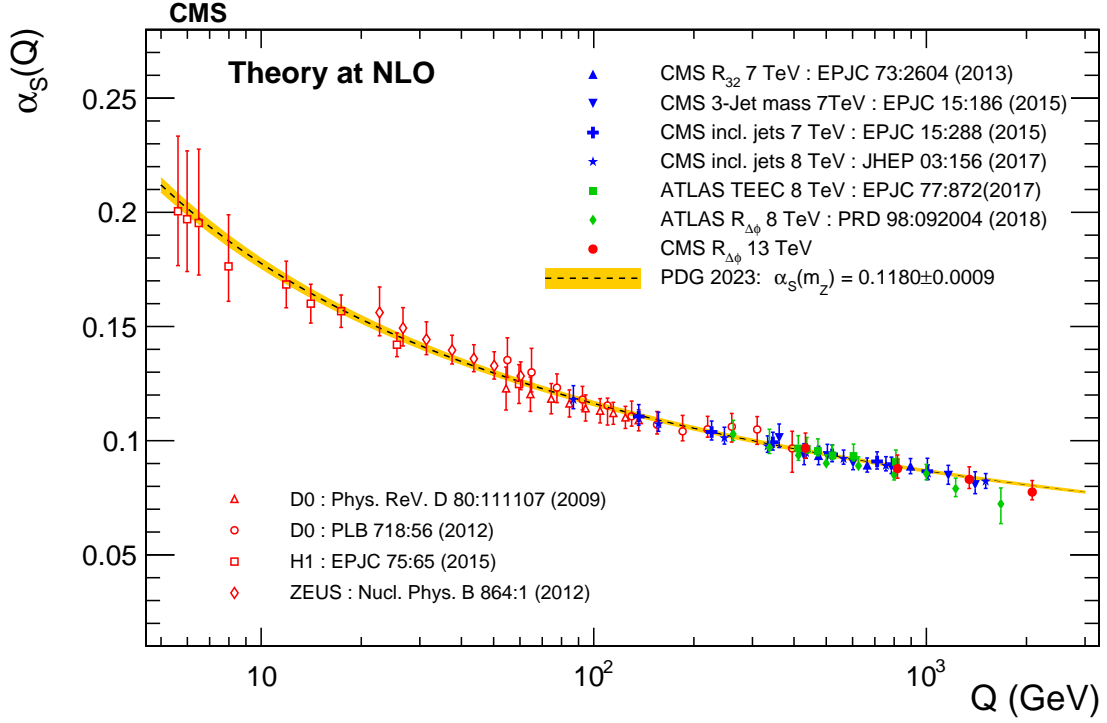


Figure 2.4.: Experimental determination by the CMS collaboration of the strong coupling α_s as a function of the scale Q . Figure taken from Ref. [39].

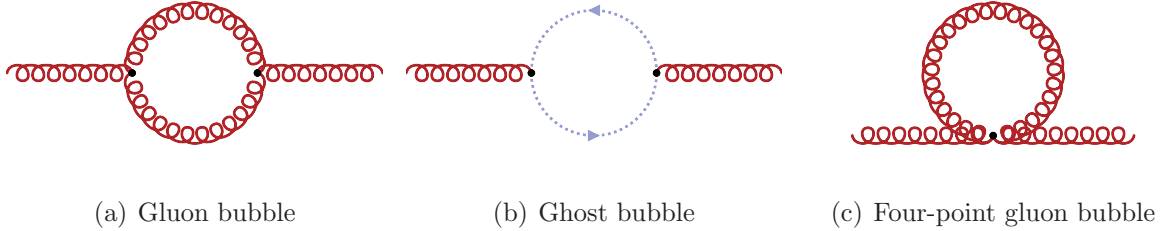


Figure 2.5.: QCD vacuum polarisation diagrams contributing to the running of α_s at the one-loop level. The light-quark loop is shown in Fig. 2.3.

hand, the strong coupling α_s varies by almost a factor of 2 from 100 GeV down to the few GeV scale. Fig. 2.4 shows the magnitude of α_s as determined by the CMS collaboration for the energy range of a few GeV up to 2 TeV [39]. The running is described by a renormalisation group equation (RGE), which for QCD takes the form [3, 38]

$$\mu_R^2 \frac{d\alpha_s}{d\mu_R^2} = \beta(\alpha_s) = -\alpha_s^2 \sum_{n=0} \alpha_s^n b_n, \quad (2.48)$$

where $\beta(\alpha_s)$ is known as the QCD β -function. The second equality is a perturbative ansatz for the β -function and the b_n are the $(n+1)$ -loop β -function coefficients. They have been computed numerically up to five loops in the $\overline{\text{MS}}$ renormalisation scheme [40–44]. The two first coefficients, b_0 and b_1 , are renormalisation-scheme independent and have the form [3]

$$b_0 = \frac{33 - 2n_f}{12\pi}, \quad b_1 = \frac{153 - 19n_f}{24\pi^2}, \quad (2.49)$$

where n_f is the number of active quark flavours contributing to the running. In Eq. (2.49) both coefficients are positive for the number of quark flavours observed in nature, which means that the QCD β -function is negative. This predicts two characteristic properties of QCD, namely that the coupling decreases at higher energies (short distances) and increases at lower energies (long distances). The former is known as asymptotic freedom and the latter predicts the formation of QCD bound states [3]. This can be seen explicitly from the solution of the RGE. At leading order it involves only b_0 , i.e. the vacuum polarisation diagrams from Fig. 2.3 and Fig. 2.5, and takes the form [21]

$$\alpha_s(\mu_R) = \frac{2\pi}{b_0} \frac{1}{\ln \frac{\mu_R}{\Lambda_{\text{QCD}}}}. \quad (2.50)$$

This means that the perturbative regime for QCD, where α_s is small enough for the expansion in Eq. (2.17) to be able to converge quickly, is the energy region above some low-energy cutoff, which is usually said to be Λ_{QCD} . In summary, perturbation theory for QCD is a well defined and useful computational framework to make predictions for processes with high enough momentum transfer, such as those at high-energy colliders.

2.3.3. Dimensional regularisation

In Sec. 2.3.1, a UV cutoff regulator was used to extract a divergence from a one-loop correction to the gluon propagator. In practice, such regulators are not used as they do not preserve gauge- and Lorentz invariance. Instead, in modern precision computations, the standard regularisation procedure is *dimensional regularisation* [45, 46]. The main reason for it being so prominent, is that calculations in this framework have turned out to be the simplest [47]. In particular, dimensional regularisation resolves divergences originating from both the ultraviolet and infrared regimes.

The mechanism of dimensional regularisation is to shift the number of space-time dimensions to $D = 4 - 2\epsilon$. The regulator ϵ is usually assumed to be real, but for purposes of analytic continuation it can also be complex. The behaviour of UV divergences is better if $\epsilon > 0$ while for IR divergences it is better to have $\epsilon < 0$. In practice, the renormalisation constants are computed first with the assumption that $\epsilon > 0$. After cancelling all UV divergences, the rest of the computation can be performed with the assumption that $\epsilon < 0$. Loop integrals in D dimensions, for example $\mathcal{M}_{\mu\nu}^{ab}$ in Eq. (2.40), are well defined and divergences manifest as poles in ϵ . After all divergences have been subtracted or cancelled, the original theory is restored by taking $\epsilon \rightarrow 0$. On a technical level, most objects and operations behave similarly when extended to D dimensions, but there are certain subtleties [21]. The action integral is D -dimensional

$$S = \int d^{d-1}x dt \mathcal{L}, \quad (2.51)$$

which necessitates $[\mathcal{L}] = D$ to preserve that $[S] = 0$. It is conventional to make parameter redefinitions such as

$$g \rightarrow \mu_R^{\frac{4-D}{2}} g, \quad (2.52)$$

to prevent the couplings from acquiring a non-integer dimensionality. Each loop thus receives a prefactor μ_R^{4-D} and the integral measure is $\int \frac{d^D k}{(2\pi)^D}$ for each loop momentum. The solid volume, which entered implicitly in Eq. (2.41), is

$$V_D = \frac{2\pi^{D/2}}{\Gamma(\frac{D}{2})}, \quad (2.53)$$

where $\Gamma(z)$ is the analytic continuation of the factorial into the complex plane. The singularity structure of solutions is usually encoded in Γ -functions. The series expansion around the regulator is $\Gamma(\epsilon) = \epsilon^{-1} + \gamma_E + \mathcal{O}(\epsilon)$ where γ_E is the Euler-Mascheroni constant.

There are various dimensional regularisation *schemes* that correspond to whether other Lorentz objects, such as Dirac matrices and metric tensors, should also be extended to D dimensions or not. The most straightforward treatment is with Conventional Dimensional Regularisation (CDR), where all Lorentz objects are treated in D dimensions. In the 't Hooft Veltman scheme (HV), *regular* vector fields, that neither appear in loops or external configurations that lead to IR divergences, are kept in four dimensions. Another class of schemes are based on Dimensional Reduction (DRED). In this case all Lorentz objects in the Feynman rules are treated strictly in four dimensions⁹ (except those that appear together with a loop momentum). In DRED regular vector fields in external states are treated in D dimensions. A closely related scheme is the Four-Dimensional Helicity scheme (FDH), where instead regular vector fields are treated in four dimensions.

The extension of the metric tensor to D dimensions is described by the following contraction and projection

$$g_{\mu\nu}g^{\mu\nu} = D, \quad g^{\mu\nu}\bar{g}_\nu{}^\rho = \bar{g}^{\mu\rho}, \quad (2.54)$$

where $\bar{g}^{\mu\rho}$ now denotes the Minkowski metric with 4-dimensional Lorentz indices. The extension of the Dirac algebra to D dimensions is in terms of a set of D four-dimensional matrices $\gamma^0, \gamma^1, \dots, \gamma^{d-1}$ that satisfy the anti-commutation relation

$$\{\gamma^\mu, \gamma^\nu\} = 2g^{\mu\nu}. \quad (2.55)$$

A well known issue in dimensional regularisation is in the definition of γ^5 . In four dimensions it is defined as $\gamma^5 = i\gamma^0\gamma^1\gamma^2\gamma^3$, which in $D = 4 - 2\epsilon$ is not compatible with preserving cyclicity of traces while also satisfying [48]

$$\{\gamma^\mu, \gamma^5\} = 0 \quad \text{and} \quad \text{Tr}\{\gamma_\mu\gamma_\nu\gamma_\rho\gamma_\delta\gamma^5\} = 4i\epsilon_{\mu\nu\rho\delta}. \quad (2.56)$$

There are thus various γ^5 -*schemes* that correspond to extensions where subsets of the above three properties are fulfilled. The most standard one is the Breitenlohner-Maison-'t Hooft-Veltman (BMHV) scheme, which gives up the anti-commutation property of Eq. (2.56) and defines γ^5 as in four dimensions. It is the most well defined and mathematically consistent scheme in the sense that it is compatible with unitarity and causality of the theory [49]. In this case we have

$$\{\hat{\gamma}^\mu, \gamma^5\} = 2\hat{\gamma}^\mu\gamma^5 \quad \text{and} \quad \{\bar{\gamma}^\mu, \gamma^5\} = 0, \quad (2.57)$$

where the Dirac matrices have been split up into a strictly 4-dimensional part $\bar{\gamma}_\mu$ and a $(D - 4)$ -dimensional part $\hat{\gamma}_\mu$, such that $\gamma_\mu = \bar{\gamma}_\mu + \hat{\gamma}_\mu$. The first relation implies $\{\hat{\gamma}^\mu, \gamma^5\} = 0$. Other options include the Larin scheme [50] and the Kreimer scheme [51]. In the former $\gamma^5 = \frac{i}{4!}\epsilon_{\mu\nu\rho\delta}\gamma^\mu\gamma^\nu\gamma^\rho\gamma^\delta$ but the anti-commutation property is dropped. In the latter we do have $\{\gamma^\mu, \gamma^5\} = 0$, but the cyclicity of traces involving an odd number of γ^5 matrices is lost. For the topics in this thesis, the choice of γ^5 -scheme has no practical relevance.

⁹Actually, in both DRED and FDH the regularising space is formally quasi-4-dimensional, and is a superset of the quasi- D -dimensional space used in CDR and HV.

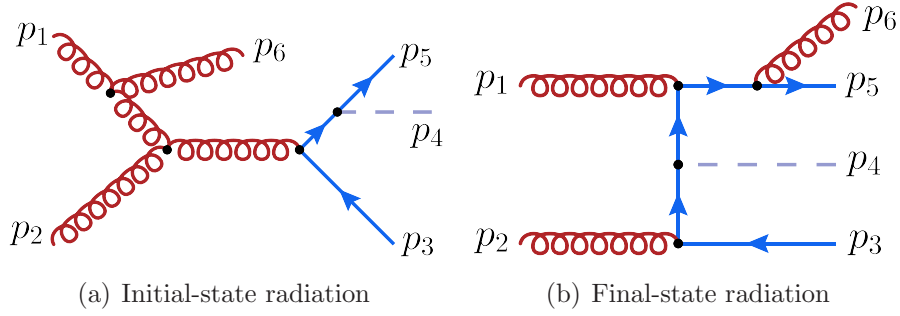


Figure 2.6.: Real radiation diagrams contributing to $gg \rightarrow t\bar{t}H$ at NLO in QCD.

2.4. Infrared divergences and cross sections

The previous section demonstrates how to extract and renormalize UV divergences, corresponding to large loop-momentum regions. If instead a loop integral is singular in the small loop-momentum regime, it is said to have an infrared (IR) divergence. Unlike UV divergences, IR divergences cannot all be resolved through parameter redefinitions, but instead cancel between contributions corresponding to different initial or final states. The reason is that additional soft and collinear radiation is indistinguishable from the hard partons of the underlying process. The Kinoshita–Lee–Nauenberg (KLN) theorem guarantees that IR divergences cancel when summing over all degenerate initial and final states [52, 53]. Therefore, when computing physical cross sections beyond LO, in addition to the so-called *virtual* loop diagrams seen earlier in this chapter, *real radiation* diagrams must be included as well. Fig. 2.6 shows examples of real radiation diagrams that contribute at NLO in $gg \rightarrow t\bar{t}H$. These are tree-level diagrams, but of order g_s^3 due to the extra gluon emission. When interfered with each other, they give contributions to the squared amplitude of the same order as 1-loop diagrams interfered with tree-level diagrams.

In general, all valid interferences that are proportional to the same power of the coupling parameter should be included in a fixed-order calculation. Fig. 2.7 illustrates various building blocks to $e^+e^- \rightarrow q\bar{q}$ up to $N^3\text{LO}$ in QCD [54]. Beyond NLO, there are mixed diagrams involving both loops of virtual particles and unresolved real radiation.

2.4.1. Cross sections

To understand how to achieve the cancellation of IR divergences between virtual and real diagrams, we consider the simplest case of a total cross section up to NLO. Following the notation of Ref. [55], an NLO cross section can very generically be expressed as

$$\sigma = \sigma_{\text{LO}} + \sigma_{\text{NLO}} = \int_n d\sigma^{\text{B}} + \int_n d\sigma^{\text{V}} + \int_{n+1} d\sigma^{\text{R}}, \quad (2.58)$$

where $d\sigma^{\text{B}}$, $d\sigma^{\text{V}}$ and $d\sigma^{\text{R}}$ are respectively the Born, virtual and real cross sections. The Born cross section is the leading-order approximation and contains no loop integrals or extra emissions. The virtual cross section includes amplitudes with IR divergent loop integrals interfered with the Born amplitude. The real contribution consists of tree-level diagrams, but with an additional emission, interfered with themselves. Due to the extra emission, the real contribution is integrated over an $(n+1)$ -particle phase space,

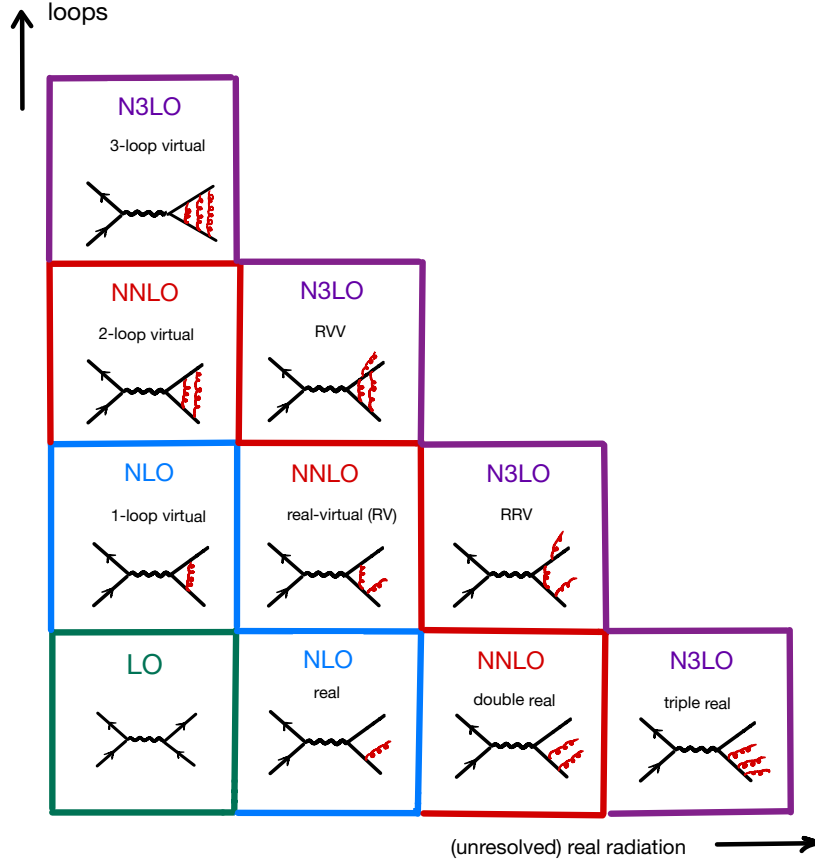


Figure 2.7.: Virtual and real amplitude ingredients contributing to $e^+e^- \rightarrow q\bar{q}$ up to N³LO in QCD. The figures for the higher-order diagrams are only representative of the type of corrections that can occur. IR singularities from virtual diagrams cancel against singularities that appear when integrating the extra real emissions over their respective subspaces. The figure is taken from Section 11 in Ref. [54].

see Eq. (2.14). The tree-level diagrams in $d\sigma^R$ are of course finite, but IR divergences appear upon integration over the phase space, from regions where the extra emission is soft and/or collinear to the hard partons.

The cancellation of IR divergences between the real and virtual contributions is complicated since the phase spaces have different multiplicity. To make the cancellation manifest, a subtraction method must be applied. At NLO, the general procedure is to include a local counterterm such that

$$\sigma_{\text{NLO}} = \int_n d\sigma^V + \int_{n+1} d\sigma^A + \int_{n+1} [d\sigma^R - d\sigma^A], \quad (2.59)$$

where $d\sigma^A$ must have the same unintegrated singular behaviour as $d\sigma^R$. By construction, the difference $d\sigma^R - d\sigma^A$ should be integrable in four dimensions such that it can be integrated numerically. Moreover, the subtraction term should be constructed such that the integration over the one-parton subspace (due to the extra emission) can be done analytically, and the IR divergences can be cancelled explicitly. In this case, the contributions to the NLO cross section can be organised as [55]

$$\sigma_{\text{NLO}} = \int_n \left[d\sigma^V + \int_1 d\sigma^A \right]_{\epsilon=0} + \int_{n+1} [(d\sigma^R)_{\epsilon=0} - (d\sigma^A)_{\epsilon=0}]. \quad (2.60)$$

Under these conditions, the remaining phase-space integrals over the resolved particles are finite in four dimensions and can be sampled and integrated with Monte Carlo techniques.

The discussion so far concerns IR divergences due to final-state radiation. There can also be IR divergences originating from collinear emissions from the initial-state partons. These are not cancelled against contributions from the virtual corrections. In processes with hadronic initial states, they are instead absorbed through redefinitions of the parton distribution functions. The general structure is to include a collinear subtraction counterterm $d\sigma^C$ such that the NLO cross section is

$$\sigma_{\text{NLO}} = \int_n d\sigma^C + \int_n \left[d\sigma^V + \int_1 d\sigma^A \right]_{\epsilon=0} + \int_{n+1} [(d\sigma^R)_{\epsilon=0} - (d\sigma^A)_{\epsilon=0}]. \quad (2.61)$$

In deep inelastic scattering, for example, the collinear counterterm is known explicitly. The contribution from a parton of type a with momentum p^μ is [55]

$$d\sigma_a^C(p) = -\frac{\alpha_s}{2\pi} \frac{1}{\Gamma(1-\epsilon)} \sum_b \int_0^1 dz \left[-\frac{1}{\epsilon} \left(\frac{4\pi\mu^2}{\mu_F^2} \right)^\epsilon P^{ab}(z) + K^{ab}(z) \right] d\sigma_b^B(zp), \quad (2.62)$$

where $P^{ab}(z)$ are the Altarelli-Parisi splitting functions [25] and $K^{ab}(z)$ determines the factorisation scheme. Similarly as to in UV renormalisation, there are various schemes corresponding to different modifications of the finite part. Taking $K^{ab}(z) = 0$ corresponds to the $\overline{\text{MS}}$ scheme.

Eq. (2.61) gives the general recipe for achieving cancellation of IR singularities, with hadronic initial states. In practice, there are many different ways of constructing the counterterm $d\sigma^A$. At NLO there are two main approaches, the FKS formalism [56–58], based on phase-space slicing, and the Catani-Seymour formalism [55, 59], based on dipole subtraction. Both are fully algorithmic, and the computation of NLO cross sections is mostly automated¹⁰. In the latter, the explicit form of the counter term, for a QCD final state, is in terms of the insertion operator $\mathbf{I}(\epsilon)$ according to [55]

$$\int_{n+1} d\sigma^A = \int_n [d\sigma^B \cdot \mathbf{I}(\epsilon)], \quad (2.63)$$

where the bracket notation refers to inserting $\mathbf{I}(\epsilon)$ into the Born squared amplitude by $_n < 1, \dots, n | \mathbf{I}(\epsilon) | 1, \dots, n >_n$. The insertion operator is defined by

$$\mathbf{I}(p_1, \dots, p_n; \epsilon) = -\frac{\alpha_s}{2\pi} \frac{1}{\Gamma(1-\epsilon)} \sum_i \frac{1}{\mathbf{T}_i^2} \mathcal{V}_i(\epsilon) \sum_{k \neq i} \mathbf{T}_i \cdot \mathbf{T}_k \left(\frac{4\pi\mu^2}{2p_i \cdot p_k} \right)^\epsilon, \quad (2.64)$$

where \mathcal{V}_i are the singular factors and \mathbf{T}_i are colour-charge operators. The sum is over the three partonic combinations, $i = gg, qg, \bar{q}q$.

Beyond NLO, specific solutions have to be tailored to different classes of processes. Examples of subtraction methods at NNLO are antenna subtraction [60, 61], q_T subtraction [62, 63], N -jettiness [64–66] and sector based methods [67, 68].

¹⁰An exception is for loop-induced processes, such as $gg \rightarrow HH$, where two-loop integrals appear already at NLO.

The redefinition of bare Lagrangian parameters in the computation of $d\sigma^V$, and the redefinition of PDFs due to $d\sigma^C$, induce a dependence on the unphysical scales μ_R and μ_F to the cross section. Therefore, fixed-order cross sections always have the dependence

$$\sigma \rightarrow \sigma(\mu_F, \mu_R) = \sigma_{\text{LO}} + \sigma_{\text{NLO}}(\mu_F, \mu_R). \quad (2.65)$$

The dependence of σ on μ_F and μ_R encodes the uncertainty due to missing higher-order terms in the perturbative expansion. Due to μ_F and μ_R being associated to factorisation and renormalisation scales, this uncertainty is referred to as a *scale uncertainty*. It is conventionally estimated by investigating how the magnitude of the cross section changes when respectively varying μ_F and/or μ_R up and down by a factor of two. Ignoring the cases when one is increased and the other is decreased, there are seven possible combinations, and the method is therefore known as the seven-point scale variation.

CHAPTER 3

Feynman Integrals

The calculation of cross sections beyond leading order requires loop amplitudes. At NLO, there are at most one-loop amplitudes that enter in the virtual cross section¹¹. The calculation of NLO cross sections, including the subtraction of infrared divergences, is fully automated for most types of processes. At NNLO, the virtual correction consists of both one-loop and two-loop amplitudes, while the real corrections contain one-loop amplitudes with an additional external leg. Fig. 2.7 shows the pattern up to N³LO for $e^+e^- \rightarrow q\bar{q}$. Calculating cross sections beyond NLO is not yet a fully automated procedure due to the presence of multi-loop amplitudes and the increased complexity of the IR subtraction and phase-space integration.

This chapter concerns the computation of multi-loop virtual amplitudes with a focus on the Feynman loop integrals. We start by introducing the basic notation by defining the momentum representation of a general Feynman integral, as well as various parametric representations, in particular the Feynman parametrisation. We then present an overview of a typical computational workflow for multi-loop amplitudes. We then review in detail the technique of integration-by-parts reduction, which has almost become a standard in modern amplitude computations. We review both analytical and numerical methods for evaluating loop integrals. In particular, the numerical framework surrounding sector decomposition, contour deformation and (quasi-)Monte Carlo integration, and their implementation in the public program PYSECDEC, is discussed in detail. Finally, in the last section we present FTINT, a program based on PYSECDEC that evaluates the closely related flow-time integrals.

¹¹The exception is for loop-induced processes, where the LO and NLO already contain one-loop and two-loop integrals respectively.

3.1. The Feynman integral

The starting point of our discussion is the momentum-space representation of a dimensionally regulated Feynman integral [47]

$$I(\nu_1, \dots, \nu_N; \{p\}) = (\mu^2)^{\nu-LD/2} \int \prod_{l=1}^L \frac{d^D k_l}{i\pi^{\frac{D}{2}}} \prod_{j=1}^N \frac{1}{P_j^{\nu_j}(\{k\}, \{p\}, m_j)}, \quad (3.1)$$

where D is the number of space-time dimensions, the $P_j^{\nu_j}(\{k\}, \{p\}, \{m_j^2\})$ are propagators corresponding to virtual particles appearing in the loops, L is the number of loops, $\{p\}$ is the set of external momenta, m_j is the particle mass of propagator j and μ is a mass parameter that restores the integer mass dimension of the D -dimensional integral. The arguments ν_i are the powers of the propagators and $\nu = \sum_i \nu_i$. Negative values correspond to terms in the numerator and $\nu_i \geq 2$ correspond to dotted propagators.

This representation of a loop integral is obtained directly after applying the momentum-space Feynman rules to a Feynman diagram, see Table 2.1. In Sec. 2.3.1 a simple example of a momentum-space loop integral appearing in QCD was shown in Eq. (2.34). There we had $L = 1$, $N = 2$ and $m_1 = m_2 = 0$, resulting in a correction to the gluon propagator

$$\mathcal{M}^{\mu\nu}(p) \sim \mu^{2\epsilon} \int \frac{d^D k}{i\pi^{\frac{D}{2}}} \frac{2k^\mu k^\nu - k^2 g^{\mu\nu} - k^\mu p^\nu - k^\nu p^\mu + g^{\mu\nu} k \cdot p}{k^2 (k-p)^2}. \quad (3.2)$$

Defining the *family* of one-loop massless bubble integrals by

$$I_{\text{bub}}(\nu_1, \nu_2; p) = (\mu^2)^{\nu-D/2} \int \frac{d^D k}{i\pi^{\frac{D}{2}}} \frac{1}{[k^2]^{\nu_1} [(k-p)^2]^{\nu_2}}, \quad (3.3)$$

we can write a very general solution ansatz as a linear combination of scalar bubble- and tadpole integrals

$$\mathcal{M}^{\mu\nu}(p) = (a_1 g^{\mu\nu} + a_2 p^\mu p^\nu) (c_1 I_{\text{bub}}(0, 1; p) + c_2 I_{\text{bub}}(1, 0; p) + c_3 I_{\text{bub}}(1, 1; p)). \quad (3.4)$$

The fact that the ansatz only contains propagator indices $\nu_i \leq 1$ is not a coincidence, but is a general feature of one-loop integrals. At one loop, all possible scalar products between loop momenta and external kinematics can be expressed as linear combinations of the basic set of propagators. Such scalar products are referred to as *reducible*. Beyond one loop, *irreducible* scalar products appear. They are the source of negative ν_i .

3.1.1. Parametric representations

The momentum representation in Eq. (3.1) is not the most convenient for evaluation. Instead, parametric representations where the integrand is rewritten as products of exponentiated polynomials are typically used. One of the most common parametrisations is by Feynman, where additional parameters are introduced according to the relation [47, 69]

$$\prod_{j=1}^N \frac{1}{P_j^{\nu_j}} = \frac{(-1)^\nu}{\Gamma_\nu} \int_0^\infty \prod_{i=1}^N dx_i x_i^{\nu_i-1} \left(\sum_{j=1}^N x_j P_j \right)^{-\nu} \delta(1 - \alpha(\mathbf{x})), \quad (3.5)$$

where we assume that the propagator power is positive. The case when $\nu_j = 0$ is trivial since P_j is effectively removed from the integral. Negative indices $\nu_j < 0$ can for example be treated by using the identity [69]

$$\frac{P_n}{\left(\sum_{i=1}^N x_j P_j\right)^{\alpha+1}} = -\frac{1}{\alpha} \frac{\partial}{\partial x_n} \frac{1}{\left(\sum_{i=1}^N x_j P_j\right)^{\alpha}}. \quad (3.6)$$

For ease of notation, we proceed here from Eq. (3.5) assuming $\nu_j \geq 0 \forall j$. The Dirac-delta function in Eq. (3.5) encodes the property of *projective invariance*. The argument $\alpha(\mathbf{x}) = \sum_i \alpha_i x_i$, where at least one $\alpha_i \neq 0$, defines a hyperplane that constrains the integral to the positive domain. The integral is invariant under different choices of hyperplanes and this freedom is sometimes referred to as the Cheng-Wu theorem [70, 71]. The integration variables are known as Feynman parameters. Inserting the relation into Eq. (3.1) yields an integral over both momenta and Feynman parameters that can be expressed as [69]

$$I(\nu_1, \dots, \nu_N; \mathbf{s}) = \frac{\Gamma(\nu)}{\prod_{j=1}^N \Gamma(\nu_j)} \int_0^\infty \prod_{j=1}^N dx_j x_j^{\nu_j-1} \delta(1 - \alpha(\mathbf{x})) \\ \times \int_{-\infty}^\infty \prod_{l=1}^L \frac{d^D k_l}{i\pi^{\frac{D}{2}}} \left[\sum_{j,i=1}^L k_j \cdot k_i M_{ji} - 2 \sum_{j=1}^L k_j \cdot Q_j + J + i\delta \right]^{-\nu}, \quad (3.7)$$

where M is an $L \times L$ matrix where each element is a sum of Feynman parameters and kinematics, Q is a vector of length L and J is a constant (with respect to the loop momenta). Shifting the loop momenta by $k_j = l_j + M_{ji}^{-1} Q_i$ the term linear in k_j is removed and the integrand is purely quadratic with respect to loop momenta. The loop momentum integration can be carried out analytically resulting in [69]

$$I(\nu_1, \dots, \nu_N; \mathbf{s}) = \frac{(-1)^\nu \Gamma(\nu - LD/2)}{\prod_{j=1}^N \Gamma(\nu_j)} \\ \times \lim_{\delta \rightarrow 0^+} \int_{\mathbb{R}_{\geq 0}^N} \prod_{j=1}^N dx_j x_j^{\nu_j-1} \frac{\mathcal{U}(\mathbf{x})^{\nu-(L+1)D/2}}{(\mathcal{F}(\mathbf{x}; \mathbf{s}) - i\delta)^{\nu-LD/2}} \delta(1 - \alpha(\mathbf{x})), \quad (3.8)$$

where $\mathcal{U}(\mathbf{x})$ and $\mathcal{F}(\mathbf{x}; \mathbf{s})$ are respectively the first and second *Symanzik polynomials* defined by

$$\mathcal{U} = \det(M), \quad \mathcal{F} = \det(M) \left[\sum_{i,j=1}^L Q_i M_{ij}^{-1} Q_j - J \right]. \quad (3.9)$$

The Symanzik polynomials are homogeneous in the Feynman parameters, $\mathbf{x} = (x_1, \dots, x_N)$, with degree L and $L+1$, respectively. This property is a requirement for the projective invariance of Feynman integrals. When applying transformations to the integrand in Eq. (3.8), it is therefore useful to try and preserve the homogeneity, if possible. We will return to this in Chapter 4.

The Symanzik polynomials can also be found from the topology of the corresponding Feynman graph. They are graph polynomials and are constructed from the spanning trees of the Feynman graph according to [47, 69]

$$\mathcal{U}(\mathbf{x}) = \sum_{T^1} \prod_{e \notin T^1} x_e, \quad \mathcal{F}_0(\mathbf{x}; \mathbf{s}) = \sum_{T^2} (-s_{T^2}) \prod_{e \notin T^2} x_e, \quad (3.10)$$

$$\mathcal{F}(\mathbf{x}; \mathbf{s}) = \mathcal{F}_0(\mathbf{x}; \mathbf{s}) + \mathcal{U}(\mathbf{x}) \sum_e m_e^2 x_e, \quad (3.11)$$

where T^1 and T^2 denote a spanning tree and spanning 2-tree respectively. A spanning tree is constructed by cutting L internal lines (edges) from the graph such that it is still a connected graph. The product of the Feynman parameters corresponding to the cut lines is a degree L monomial, at most linear in each parameter. The \mathcal{U} polynomial is the sum of all such monomials obtained from the possible spanning trees of the associated graph. For example, if x_1 and x_2 are the Feynman parameters corresponding to the propagators of the bubble integral, $\mathcal{U} = x_1 + x_2$ since there are only two possible cuts of one line each that leaves the bubble connected.

A spanning 2-tree is constructed by cutting one additional internal line, that is $L + 1$ cuts for a graph with L loops, which results in a disconnected graph. The product of the Feynman parameters corresponding to the cut edges constitute a degree $L + 1$ monomial. The $\mathcal{F}_0(\mathbf{x}; \mathbf{s})$ is the sum of all such monomials multiplied by the respective $-s_{T^2}$, corresponding to all possible T^2 , where the symbol $s_{T^2} = (\sum_{j \in \text{Cut}(T^2)} p_j)^2$ denotes the square of the momentum flowing between the cut of the components of the 2-tree T^2 . Finally, $\mathcal{F}(\mathbf{x}; \mathbf{s})$ is constructed by adding to $\mathcal{F}_0(\mathbf{x}; \mathbf{s})$, the \mathcal{U} polynomial multiplied by the sum over all Feynman parameters times the mass of the corresponding propagator. For massless integrals, the second term does not contribute and $\mathcal{F}(\mathbf{x}; \mathbf{s}) = \mathcal{F}_0(\mathbf{x}; \mathbf{s})$. For example, for the massless bubble integral there is only one possible spanning 2-tree and the second Symanzik polynomial is therefore $\mathcal{F}(\mathbf{x}; \mathbf{s}) = -sx_1x_2$. Inserting the explicit $\mathcal{U}(\mathbf{x})$ and $\mathcal{F}(\mathbf{x}; \mathbf{s})$ for the one-loop bubble into Eq. (3.8) yields the Feynman representation of the massless bubble integral as

$$I_{\text{bub}, m=0}(s) = \lim_{\delta \rightarrow 0^+} \Gamma(\epsilon) \int_{\mathbb{R}_{\geq 0}^2} dx_1 dx_2 \frac{(x_1 + x_2)^{-2+2\epsilon}}{(-sx_1x_2 - i\delta)^\epsilon} \delta(1 - x_1 - x_2), \quad (3.12)$$

where the arbitrary (but common¹²) choice $\delta(1 - \alpha_1x_1 - \alpha_2x_2) = \delta(1 - x_1 - x_2)$ is made. With massive propagators the \mathcal{F} polynomial receives additional contributions from the second term and in this case (with $m = m_1 = m_2$) $\mathcal{F}(\mathbf{x}; \mathbf{s}) = -sx_1x_2 + m^2(x_1 + x_2)^2$ and the Feynman representation of the massive bubble integral is

$$I_{\text{bub}, m \neq 0}(s) = \lim_{\delta \rightarrow 0^+} \Gamma(\epsilon) \int_{\mathbb{R}_{\geq 0}^2} dx_1 dx_2 \frac{(x_1 + x_2)^{-2+2\epsilon}}{(-sx_1x_2 + m^2(x_1 + x_2)^2 - i\delta)^\epsilon} \delta(1 - x_1 - x_2). \quad (3.13)$$

This demonstrates a characteristic feature of massive Feynman integrals, namely the appearance of *quadratic* integration variables in the \mathcal{F} polynomial. The \mathcal{F} polynomials of massless Feynman integrals are always at most linear in each Feynman parameter.

¹²Since for 1-loop integrals this results in setting $\mathcal{U} = 1$.

While the Feynman parametrisation is the most common for evaluating loop integrals, there are other parametrisations that are useful for various applications. An almost equivalent parametrisation, that we use in Sec. 3.4, is the Schwinger parametrisation. It is derived in almost the same way as the Feynman parametrisation, but instead we use the identity [69]

$$\frac{1}{P_j^{\nu_j}} = \frac{(-1)^{\nu_j}}{\Gamma(\nu_j)} \int_0^\infty d\alpha \alpha^{\nu_j-1} e^{\alpha P_j}, \quad \nu_j > 0, \quad (3.14)$$

where as before we assume that the propagator power is positive. Negative powers can again be treated by the identity in Eq. (3.6). Inserting Eq. (3.14) into the momentum representation in Eq. (3.1) for each propagator we get

$$I(\nu_1, \dots, \nu_N; \mathbf{s}) = \prod_{l=1}^L \int \frac{d^D k_l}{i\pi^{\frac{D}{2}}} \prod_{j=1}^N \frac{(-1)^{\nu_j}}{\Gamma(\nu_j)} \int_0^\infty d\alpha_j \alpha_j^{\nu_j-1} \exp\left(\sum_{k=1}^N \alpha_k D_k\right). \quad (3.15)$$

Now the same shift in the loop momentum as before can be applied to obtain a quadratic form, which allows for Gaussian integration over the loop momentum. The result for the Schwinger parametrisation is the following representation

$$I(\nu_1, \dots, \nu_N; \mathbf{s}) = (-1)^\nu \prod_{j=1}^N \left(\frac{1}{\Gamma(\nu_j)} \int_0^\infty d\alpha_j \alpha_j^{\nu_j-1} \right) \mathcal{U}^{-D/2} \exp\left(-\frac{\mathcal{F}}{\mathcal{U}}\right), \quad (3.16)$$

where \mathcal{U} and \mathcal{F} are again the first and second Symanzik polynomials defined in Eq. (3.9). It is possible to retrieve the Feynman parametrisation from Eq. (3.16) by inserting a carefully chosen identity [47]

$$1 = \int_0^\infty dt \delta\left(t - \sum_{j=1}^N \alpha_j\right) \quad (3.17)$$

and making a change of variables $x_j = \alpha_j/t$.

Finally, we introduce without derivation two more parametrisations. The first is the Lee-Pomeransky representation, given by [69, 72]

$$I(\nu_1, \dots, \nu_N; \mathbf{s}) = \frac{(-1)^\nu \Gamma(D/2)}{\Gamma((L+1) D/2 - \nu) \prod_j \Gamma(\nu_j)} \int_0^\infty \prod_{j=1}^N dz_j z_j^{\nu_j-1} (\mathcal{U} + \mathcal{F})^{-D/2}. \quad (3.18)$$

Starting from Eq. (3.18), the Feynman parametrisation can again be obtained by inserting the identity in Eq. (3.17), with z_j instead of α_j , and then making the same type of variable change $x_j = z_j/t$. The advantage of the Lee-Pomeransky representation is that the graph polynomials are inside the same bracket, which can make certain geometric interpretations more convenient. The second and last parametrisation we introduce is the Mellin-Barnes (MB) representation. This representation is special since for loop integrals, we first start from another parametric representation, and only then generate the $(n-1)$ -fold MB integrals via the following formula [73]

$$\begin{aligned} \frac{1}{(X_1 + \dots + X_n)^\lambda} &= \frac{1}{\Gamma(\lambda)} \frac{1}{(2\pi i)^{n-1}} \int_{-i\infty}^{+i\infty} \dots \int_{-i\infty}^{+i\infty} dz_2 \dots dz_n \prod_{i=2}^n X_i^{z_i} \\ &\times X_1^{-\lambda-z_2-\dots-z_n} \Gamma(\lambda + z_2 + \dots + z_n) \prod_{i=2}^n \Gamma(-z_i). \end{aligned} \quad (3.19)$$

If the starting point was the Feynman representation, for example, the X_i would be the terms in the \mathcal{F} polynomial. In that case, the exponent λ would be the exponent of \mathcal{F} that regulates the integral. The multiple MB integrals that appear on the right-hand side can be evaluated either numerically or analytically, depending on the problem.

3.1.2. Divergences

It is shown in Chapter 2 how loop integrals contain UV and IR divergences associated to the behaviour of the integral at large respectively small (or collinear) loop momenta. In the Feynman representation the UV divergences are encoded in the Γ -function in the prefactor, as well as in $\mathcal{U}(\mathbf{x})$ in the numerator. For example, the bubble integral in Eq. (3.13) contains a $\Gamma(\epsilon)$. Expanding around $\epsilon = 0$ we identify the expected UV divergence as a pole in ϵ since

$$\Gamma(\epsilon) = \frac{1}{\epsilon} + \gamma_E + \mathcal{O}(\epsilon). \quad (3.20)$$

The IR divergences are encoded in the exponents of the monomials in $\mathcal{F}(\mathbf{x})$ ¹³. Necessary, but not sufficient, conditions to have an IR singularity are given by the Landau equations [76–79]. They can be formulated in both momentum and parametric representations. In the Feynman parametrisation they take the form [80]

$$\mathcal{F}(\mathbf{x}; \mathbf{s}) = 0, \quad x_k \frac{\partial \mathcal{F}(\mathbf{x}; \mathbf{s})}{\partial x_k} = 0, \quad \text{for all } k \in \{1, \dots, N\}. \quad (3.21)$$

These are not sufficient conditions for an IR pole since the power of $\mathcal{F}(\mathbf{x}; \mathbf{s})$ could be such that it appears in the numerator. Note that the first condition follows from the second one by Euler’s homogeneous function theorem

$$\sum_{k=1}^N x_k \frac{\partial \mathcal{F}(\mathbf{x}; \mathbf{s})}{\partial x_k} = (L + 1) \mathcal{F}(\mathbf{x}; \mathbf{s}). \quad (3.22)$$

We thus have a singularity if for all k , either x_k , or the derivative $\frac{\partial \mathcal{F}(\mathbf{x}; \mathbf{s})}{\partial x_k}$, vanishes. Singularities corresponding to $x_k \rightarrow 0$ are located at the boundary of integration and can be extracted algorithmically with sector decomposition, for example. Singularities corresponding to vanishing derivatives instead appear *inside* the integration domain and are known as *pinch singularities*. This can happen when monomials in $\mathcal{F}(\mathbf{x}; \mathbf{s})$ have coefficients of different sign and cancel against each other [81]. A solution of Eq. (3.21) with all $x_k \neq 0$ is referred to as a *leading* Landau singularity.

A special case which is particularly relevant to Chapter 4, is when only $\mathcal{F}(\mathbf{x}; \mathbf{s})$ vanish (and not necessarily its derivatives) away from the boundary of integration. As long as $x_k \frac{\partial \mathcal{F}(\mathbf{x}; \mathbf{s})}{\partial x_k} \neq 0$ for some k , the Landau equations tell us that this cannot correspond to physical IR singularities. Instead, these are integrable singularities. They can be treated with contour deformation, for example, which resolves the singularities at the cost of increased computational complexity. The mechanism of contour deformation is to shift each integration variable by a small imaginary part $i\tau_k$

$$x_k \rightarrow z_k = x_k - i\tau_k, \quad (3.23)$$

¹³In parametric representations, divergences can also be found by considering the scaling behaviour of the integrand close to the integration boundary, with respect to different integration parameters [74, 75].

which results in the \mathcal{F} -polynomial transforming as

$$\mathcal{F}(\mathbf{x}; \mathbf{s}) \rightarrow \mathcal{F}(\mathbf{z}; \mathbf{s}) = \mathcal{F}(\mathbf{x}; \mathbf{s}) - i \sum_k \tau_k \frac{\partial \mathcal{F}(\mathbf{x}; \mathbf{s})}{\partial x_k} + \mathcal{O}(\tau^2). \quad (3.24)$$

The individual τ_k should be chosen in accordance with the causal $i\delta$ prescription, i.e. to ensure that the deformed $\mathcal{F}(\mathbf{z}; \mathbf{s})$ develops a negative imaginary part where $\mathcal{F}(\mathbf{x}; \mathbf{s})$ vanishes. If the τ_k are sufficiently small, the original and deformed contours can be connected without enclosing any poles, ensuring by Cauchy's theorem that the deformation leaves the integral invariant. An example of a choice of shift parameters that achieves all of the above (in the integration domain $x_i \in [0, 1]$) is given by [82–94],

$$\tau_k = \lambda_k x_k (1 - x_k) \frac{\partial \mathcal{F}(\mathbf{x})}{\partial x_k}, \quad (3.25)$$

where λ_k are arbitrary parameters chosen small enough that the loop does not enclose any poles. Inserting the deformation into Eq. (3.24) gives,

$$\mathcal{F}(\mathbf{z}; \mathbf{s}) = \mathcal{F}(\mathbf{x}; \mathbf{s}) - i \sum_k \lambda_k x_k (1 - x_k) \left(\frac{\partial \mathcal{F}(\mathbf{x}; \mathbf{s})}{\partial x_k} \right)^2 + \mathcal{O}(\tau^2). \quad (3.26)$$

For sufficiently small λ_k , such that we can neglect terms of $\mathcal{O}(\tau^3)$ which contribute to the imaginary part with a positive sign, this choice gives a negative imaginary part to $\mathcal{F}(\mathbf{z}; \mathbf{s})$ except at $x_k = 0$ and $x_k = 1$ (chosen as the boundary of integration) and where all $\partial \mathcal{F}(\mathbf{x}; \mathbf{s}) / \partial x_k$ vanish.

In Chapter 4 an alternative strategy to resolve integrable singularities is presented, that avoids the main computational drawbacks associated with contour deformation.

3.2. Computing amplitudes

The computation of amplitudes beyond the one-loop level is far from an automated procedure. Instead, solutions are usually tailored to specific processes. Nevertheless, over time certain techniques and frameworks have proven particularly useful, and a certain standard workflow in multi-loop amplitude computations has taken form. A common breakdown of the computational workflow is in three main steps as follows:

1. Amplitude generation
2. Reduction to master integrals
3. Integral evaluation

We provide a summary of the main steps here, and describe in detail specific methods for the last two steps in the later sections.

Generation

The first step is conceptually complicated but is in many cases not a computational bottleneck. It involves the generation of all contributing Feynman diagrams, the translation to amplitude expressions via application of Feynman rules and performing tensor

algebra related to, for example, the spinor and colour structures of the diagram. In Sec. 2.2.3, most of these steps were performed explicitly for parts of the leading-order $gg \rightarrow t\bar{t}H$ amplitude. At higher orders in perturbation theory, the generation of diagrams as well as colour and spinor traces become too tedious to do by hand, but are often fast enough on a computer. Compared to the extreme computational complexity of multi-loop Feynman integrals, the generation of the amplitude is not considered to make up any significant bottleneck in the computational workflow. Still, there are certain choices of how to *represent* the amplitude that can affect the computational complexity of the later steps. A standard strategy is the method of projectors, where the amplitude is written as a linear combination of certain allowed tensor structures, and the coefficients are referred to as form factors. Approaches using so-called physical projectors, that single out helicity amplitudes involving only four-dimensional external states, get rid of spurious tensor structures and have proven to sometimes yield much smaller coefficients than the traditional approach [95, 96]. Another more straightforward possibility is to use the tree-level amplitude as a projector and interfere it with the virtual amplitudes. This obviously only works for non loop-induced processes.

The most commonly used programs to generate the relevant Feynman diagrams for a given process is QGRAF [97] and FEYNARTS [98], and a recently released alternative is FEYNGRAPH [99]. To perform the tensor algebra, the most performant and widely used program is FORM [100], where also COLOR.H [101] is implemented. There exists various programs that, among other things, stitch together the above steps to partially automate the generation of amplitudes. Such programs include ALIBRARY [102], FEYNALC [103–105] and TAPIR [106].

Reduction

After the amplitude is generated the remaining task is to evaluate the loop integrals. In a way, the last two steps can be considered two sides of the same coin. It is in principle possible to parametrise the integrals as they appear directly in the amplitude and evaluate them right away, but this quickly becomes computationally unfeasible with increasing number of loops and legs. Fortunately, the integrals that appear inside the graphs of an amplitude are not all independent. They form a vector space, and a significant reduction of computational complexity can be achieved by projecting the amplitude onto independent basis elements of this space, referred to as *master integrals*. Such a procedure is known as amplitude *reduction*. Beyond one loop, this is almost always *integration-by-parts* (IBP) reduction [107, 108]. The result is a representation of the amplitude as a linear combination of master integrals. Besides reducing the number of integrals that needs to be evaluated, the reduction allows choosing which master integrals should be used. Which master integrals to pick depends on the problem and which tool is used to evaluate the integrals in the end.

Evaluation

The final step is to evaluate the chosen master integrals and compute the amplitude. It is usually only at this stage that a parametrisation is introduced¹⁴. Evaluating the

¹⁴Recent ideas for also performing the IBP reduction in parameter space have been introduced in [109].

integrals can be done either analytically or numerically. If an analytic solution is available this is the preferred option¹⁵, but in many cases the required number of loops, legs and kinematic scales means that this is not the case and instead one has to resort to numerical strategies. Evaluating the integral numerically means substituting numbers for all kinematics and repeatedly performing the integration over the parameters, usually with some Monte Carlo (MC) method. This allows computing the amplitude at a finite set of kinematic points. Whether this set of points is enough, depends on how many points are required by the Monte Carlo program calculating some observable (see for example the phase-space integrals in Eq. (2.61)). A common strategy is to evaluate the loop amplitude on a fixed phase-space grid and interpolate for the values in between. This requires getting a good handle on the grid uncertainties and understanding how they propagate to observables, which are the only physically interesting quantities in the end. This is the topic of Chapter 5.

3.2.1. IBP reduction

Relations between dimensionally regulated Feynman integrals appearing in a loop amplitude can be found through so-called IBP relations. They are derived starting from the fact that in dimensional regularisation the integral over a total derivative vanishes [47, 107, 108, 110]. For a loop integral this means

$$\int \prod_{l=1}^L \frac{d^D k_l}{i\pi^{\frac{D}{2}}} \frac{\partial}{\partial k_i^\mu} q^\mu \prod_{j=1}^N \frac{1}{P_j^{\nu_j}(\{k\}, \{p\}, m_j)} = 0, \quad i = 1, \dots, L, \quad (3.27)$$

where q^μ is a linear combination of external and loop momenta. Selecting various such combinations yields relations between different members of the same *integral family*. A family is defined as the set of integrals with the same propagators, including possible auxiliary propagators due to irreducible scalar products, that differ in the indices ν_j . A propagator P_j is dotted if $\nu_j > 1$ and pinched if $\nu_j = 0$.

As an example we return to the bubble integral. The family of one-loop massless bubble integrals is defined in Eq. (3.3). Adding a mass m , the family of equal-mass bubbles is

$$I_{\text{bub}}(\nu_1, \nu_2; p, m) = (\mu_R^2)^{\nu-D/2} \int \frac{d^D k}{i\pi^{\frac{D}{2}}} \frac{1}{[k^2 - m^2]^{\nu_1} [(k-p)^2 - m^2]^{\nu_2}}, \quad (3.28)$$

Now we use Eq. (3.27) with $q^\mu = p^\mu$ and obtain

$$\begin{aligned} 0 &= p^\mu (\mu^2)^{\nu-D/2} \int \frac{d^D k}{i\pi^{\frac{D}{2}}} \frac{\partial}{\partial k^\mu} \frac{1}{[k^2 - m^2]^{\nu_1} [(k-p)^2 - m^2]^{\nu_2}} \\ &= (\mu^2)^{\nu-D/2} \int \frac{d^D k}{i\pi^{\frac{D}{2}}} \left[\frac{-2\nu_1 (p \cdot k)}{[k^2 - m^2]^{\nu_1+1} [(k-p)^2 - m^2]^{\nu_2}} + \frac{-2\nu_2 (p^2 - p \cdot k)}{[k^2 - m^2]^{\nu_1} [(k-p)^2 - m^2]^{\nu_2+1}} \right] \\ &= (\nu_1 - \nu_2) I_{\nu_1 \nu_2} + \nu_1 I_{\nu_1+1, \nu_2-1} + \nu_2 I_{\nu_1-1, \nu_2+1} + \nu_2 I_{\nu_1-1, \nu_2+1} + \nu_1 p^2 I_{\nu_1+1, \nu_2} - \nu_2 p^2 I_{\nu_1, \nu_2+1}, \end{aligned} \quad (3.29)$$

where on the last line the notation $I(\nu_1, \nu_2; p, m) = I_{\nu_1, \nu_2}$ has been introduced. The last line is one so-called IBP relation. It relates dotted and pinched bubbles, that is,

¹⁵A caveat is that obtaining numbers from the special functions in the analytical solutions is not always fast.

different integrals in the bubble family, to each other. Just one IBP relation in itself is not very useful. Repeating the same procedure with $q^\mu = k^\mu$ and then considering certain linear combinations of the two IBP identities, provides a system that can be used to reduce any I_{ν_1, ν_2} to a linear combination of I_{11} and I_{10} [47]. We say that I_{11} and I_{10} are master integrals of the equal-mass bubble family¹⁶.

In addition to the IBP identities derived above, there are also relations stemming from so-called Lorentz-invariance (LI) identities [111]

$$\sum_{i=1}^E \left(p_i^\nu \frac{\partial}{\partial p_{i,\mu}} - p_i^\mu \frac{\partial}{\partial p_{i,\nu}} \right) I_{\nu_1, \dots, \nu_N}, \quad (3.30)$$

where E is the number of external legs. The LI identities have been shown to not provide any additional linearly independent relations [112], but they are known to accelerate the reduction procedure and are thus still used by IBP reduction programs.

A systematic method to construct IBP systems is with the Laporta algorithm [113]. The algorithm consists of using identities, such as Eq. (3.29) for the bubble integral, with different choices of the indices ν_j , to generate a system of linear IBP equations that can be solved with Gaussian elimination. Since in more complicated cases, the system is usually overdetermined, a criterion for which integrals to pick is required. This is how the freedom of picking a particular set of master integrals arises. The generation procedure, known as *seeding*, is non-trivial since arbitrary choices of, for example, the maximum values of the indices ν_j have to be made. Optimizing the seeding procedure is an active research field and recent developments can be seen in for example Ref. [114].

Modern IBP programs utilise finite field and interpolation techniques to accelerate solving complicated IBP systems [115–120]. This alleviates memory issues associated to solving large systems of linear equations symbolically. In some cases, the systems are still too complicated to be reconstructed analytically and one has to resort to pure numerical solutions. In this case the IBP system is generated once and then repeatedly solved numerically for each required kinematic point.

Popular programs that generate and solve IBP systems include, but are not limited to, KIRA [114, 121, 122], FIRE [115, 123] and REDUZE [124, 125]. It was shown with RATRACER [126] that a significant speed-up can be obtained by recording the arithmetic operations of an IBP solution (in the context of finite-field methods) as an execution trace that is replayed for each required IBP probe.

Basis selection

The most relevant aspect of IBP reduction for the topics in this thesis is the freedom of picking which master integrals to express the amplitude in. A set of such master integrals is known as an IBP basis. A list of good properties to aim for are [16]

1. Quasi-finite master integrals [75]
2. D -factorising IBP coefficients [127–129]
3. Fast-to-evaluate master integrals

¹⁶The IBP relations lead to three master integrals I_{10} , I_{01} and I_{11} , which are reduced to two due to the equal-mass symmetry relation $I_{\nu 0} = I_{0\nu}$ [47].

4. Simple denominators in the IBP coefficient

A quasi-finite basis is achieved by extracting IR and UV singularities from the master integrals into the IBP coefficients. Since the IR and UV singularities cannot depend on the kinematic variables, it must always be possible to achieve a D -factorised form of the IBP coefficient. To obtain a quasi-finite basis with D -factorising IBP coefficients, dimension shifts ($D = 6 - 2\epsilon$, $D = 8 - 2\epsilon$, etc.) and dotted propagators of the master integrals are considered.

Quasi-finite integrals are known to be convenient for numerical evaluation of master integrals. Aside from this, different bases may lead to faster evaluation depending on what specific tool is used. Here heuristics, together with an iterative selection strategy, are often used. A typical workflow is to ensure the first two properties, then benchmark which integrals in the D -factorising, quasi-finite basis evaluate quickly. Integrals that show undesired features, such as being too slow to evaluate, are discarded. The procedure is then repeated, but the construction of the initial basis is restricted to not use the excluded integrals from the previous iterations. The ability to pick specific integrals to build an amplitude is a useful advantage to keep in mind for the topics of Chapter 4.

3.2.2. Integral evaluation

There are two main approaches in the evaluation of master integrals. Either one tries to solve them analytically while keeping the kinematics symbolic, or one inserts numbers for each kinematic variable and integrates over the parameters numerically. There exists a huge number of methods on both sides, each with their own advantages and disadvantages. Also, it is possible to use a mixture of numerical and analytical evaluation by, for example, integrating out one Feynman parameter symbolically and doing the rest numerically [84, 130]. Generally, if possible, an analytic solution is preferred since then the integration does not have to be repeated for each phase-space point. It also makes the treatment of singularities more straightforward. For example, the cancellation of IR poles can be made exact, while in a numerical solution poles only cancel up to the numerical uncertainty. On the other hand, with increasing number of loops and kinematic scales, analytic solutions eventually become intractable. This problem can to an extent be alleviated by using expansions, see below. Moreover, automation is much more difficult in the analytic approach, since the underlying function classes change with increasing loop order [69].

Analytical methods

Evaluating Feynman integrals analytically has almost become synonymous with using the method of differential equations (DE) [111, 131, 132]. The reason is due to the insight of writing the system on ϵ -factorised form [133]

$$\frac{\partial}{\partial s_i} \vec{I}(D, s_i) = \epsilon A(s_i) \vec{I}(D, s_i), \quad (3.31)$$

where s_i is a kinematic invariant, D is the dimension and $A(s_i)$ is the basis matrix of IBP coefficients. When the ϵ -dependence is successfully factorised from the basis

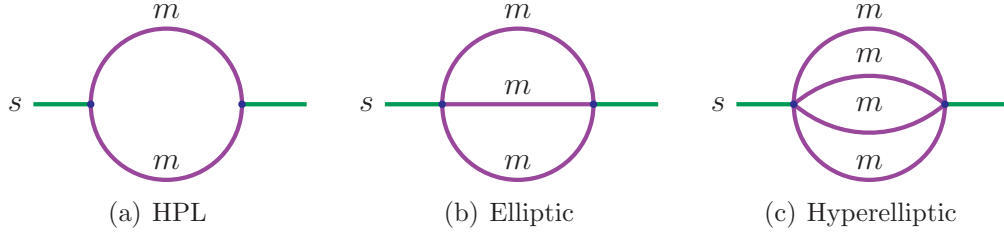


Figure 3.1.: The three first members of the equal-mass banana family, usually referred to as the bubble, sunrise and banana integrals respectively. The geometry of the L -loop banana integral (for $L \geq 2$) is described by a Calabi Yau $(L - 1)$ -fold. At two loops there is an elliptic curve and a three loops there is a $K3$ surface [137].

matrix, the system is said to be on canonical form. The equation above is usually written as [69]

$$d\vec{I}(\epsilon, s) = \epsilon \left(d\tilde{A} \right) \vec{I}(\epsilon, s), \quad \tilde{A} = \sum_k A_k \ln \alpha_k(s), \quad (3.32)$$

where $\alpha_k = s - s_k$ are letters that define the location of the singularities. In this form the equation has a solution in terms of iterated integrals [69]

$$\vec{I}(\epsilon, s) = \left(1 + \epsilon \int^s dt_1 \tilde{A}(t_1) + \epsilon^2 \int^s dt_1 \int^{t_1} dt_2 \tilde{A}(t_1) \tilde{A}(t_2) + \dots \right) \vec{I}_0(\epsilon). \quad (3.33)$$

If the letters α_k are rational functions the solution can be expressed in terms of multiple polylogarithms (MPLs) [134, 135], defined iteratively by

$$\begin{aligned} G(0, \dots, 0; z) &= \frac{1}{n!} \ln^n z, \quad G(; z) = 1, \\ G(a_1, \dots, a_n; z) &= \int_0^z \frac{dt}{t - a_1} G(a_2, \dots, a_n; t), \end{aligned} \quad (3.34)$$

where n is the *weight*. If $a_1 \in \{-1, 0, 1\}$ they are referred to as harmonic polylogarithms (HPLs) [136].

The method of differential equations (from the analytical side) is particularly competitive when the canonical form can be achieved, and when the solution can be written in terms of MPLs. However, with increasing number of kinematic scales, such as internal masses, the solution often goes beyond MPLs into less explored function territory. For example, the two-loop banana integral is known to be a so-called *elliptic integral*. An elliptic integral is characterised by an at least cubic polynomial appearing under a square root in the integrand. The three-loop banana integral with equal masses is related to a Calabi-Yau 2-fold [137] and is sometimes referred to as hyperelliptic. Fig. 3.1 illustrates the increasing complexity of the underlying geometry describing the three first members of the equal-mass banana family. The issue with such integrals is that it is usually very challenging to obtain the ϵ -factorised form, and it is not yet known if it exists in all cases¹⁷.

¹⁷Actually, in the particular case of the equal-mass banana family, a systematic method of obtaining the ϵ -factorised form was presented in Ref. [137]. For higher point functions, and if the masses are not equal, the situation quickly becomes more complicated.

The stagnation of analytical methods for multi-loop problems with many masses is a strong motivation for the development of numerical strategies, as they are usually not sensitive to the increasing complexity of the function class of the solution space.

Numerical methods

While analytical solutions of Feynman integrals often involve differential equations, in the field of numerical evaluations there is a much wider choice of methods. We briefly summarise a few popular ones here. A detailed description of the method of sector decomposition, with related tools, is given in Sec. 3.3.

The DE method described in the analytical section can also be applied by solving the differential equations numerically [67, 138, 139]. Due to the presence of kinematic singularities, the integration contour must be deformed into the complex plane to retain good convergence. The boundary conditions can be evaluated with sector decomposition, for example, or by considering the integral in an expansion.

A method which does not require analytic reductions is sector decomposition [140–142]. Sector decomposition algorithmically isolates and subtracts overlapping endpoint singularities of dimensionally regulated parameter integrals¹⁸. The result is a Laurent expansion in the dimensional regulator ϵ

$$I = \sum_{m=-r}^{2L} \frac{C_m}{\epsilon^m} + \mathcal{O}(\epsilon^{r+1}), \quad (3.35)$$

where the coefficients C_m are finite parameter integrals that can be evaluated with MC integration, for example. The basic idea can be demonstrated with the following two-dimensional integral [69]

$$I = \int_0^1 dx \int_0^1 dy (x+y)^{-2+\epsilon}. \quad (3.36)$$

The integral has a singular point as both $x, y \rightarrow 0$ simultaneously. Sector decomposition resolves the singularity by splitting the integral into *sectors*

$$I = \int_0^1 dx \int_0^1 dy (x+y)^{-2+\epsilon} [\Theta(x-y) + \Theta(y-x)], \quad (3.37)$$

where $\Theta(x)$ is the Heaviside step function. Making the replacement $y = xt$ in the first integral and $x = yt$ in the second integral, the split integration region is remapped to the unit square. The final result is [142]

$$I = 2 \int_0^1 dx^{-1+\epsilon} \int_0^1 dt (1+t)^{-2+\epsilon}. \quad (3.38)$$

The overlapping singularity is now factorised and the divergence is made explicit in the exponent of $x^{-1+\epsilon}$; the integral diverges logarithmically. In this case we can immediately

¹⁸These are usually Feynman loop integrals but other applications exist. In Sec. 3.4 the closely related flow-time integrals appearing within the gradient-flow formalism, and their computation with sector decomposition, is described. It has also previously been applied to phase-space integrals [143–147].

identify the integral over x as $B(\epsilon, 1)$, where $B(z_1, z_2)$ is the Euler-Beta function defined as

$$B(z_1, z_2) = \int_0^1 t^{z_1-1} (1-t)^{z_2-1} dt = \frac{\Gamma(z_1)\Gamma(z_2)}{\Gamma(z_1+z_2)}. \quad (3.39)$$

Using the properties of the Γ function we get the divergent part explicitly as

$$\frac{\Gamma(\epsilon)}{\Gamma(1+\epsilon)} = \frac{\Gamma(\epsilon)}{\epsilon\Gamma(\epsilon)} = \frac{1}{\epsilon} \quad (3.40)$$

and we see how the Laurent expansion arises. In more complicated cases the singularities result from re-added subtraction terms that are introduced to regulate the isolated endpoint divergences. The method shown here is known as *iterated* sector decomposition, since in more complicated cases with more variables, the procedure has to be repeated until the sector decomposed form is reached. In modern programs *geometric* sector decomposition, based on convex geometry, is the favoured strategy as it usually leads to fewer sectors. Geometric sector decomposition as implemented in PYSECDEC is described in Sec. 3.3.

Other methods include direct computation [148–150], loop-tree duality [151–157] and MB integration [158]. Direct computation methods utilize a finite-width regulator to resolve kinematic singularities and improve numerical convergence. If the regulator is sufficiently small the resulting systematic uncertainty is negligible compared to the statistical uncertainty of the MC integration. A problem arises if no value of the regulator can be found that both ensures a small systematic error while also keeping the numerical convergence stable enough. In Refs. [159, 160] direct computation is used together with Richardson extrapolation [161], which results in stable results for specific choices of $\alpha(\mathbf{x})$ in Eq. (3.8). Loop-tree duality is a method that works in four-dimensions, that is, it avoids dimensional regularisation. The idea is to achieve local cancellation of singularities at the integrand level, by combining virtual and real contributions under one integral [162]. It achieves this by transforming the integration domain to Euclidean space where there is better intuition for the singular structure. Loop-tree duality effectively reduces the dimension of the integrals and therefore scales very well with the loop order. Finally, numerical MB integration utilises the MB representation (see Eq. (3.19)) to rewrite parametric integrals as multiple MB integrals, that are then evaluated numerically.

Expansion methods

A different approach compared to the methods described above are those based on expansions. The most well known is expansion by regions (EBR) [31] which, like sector decomposition, can be derived in the geometric interpretation [163–165]. The idea is to expand the integral into regions based on the scaling properties of some smallness parameter, where the integration is simpler as it involves fewer kinematic scales. In each limit the simplified integral can then be computed either numerically or analytically.

The obvious drawback of using an expansion is that the result may only be valid in a certain region of the parameter space. A popular approach is therefore to combine various expansions, to cover as much of the parameter space as possible. For example, combining a high-energy expansion with a small p_T expansion, has been shown to yield accurate results in all regions of the phase space for a variety of observables in

$gg \rightarrow HZ, HH$ [166, 167]. A similar approach is used in Ref. [168], where the low p_T expansion is instead replaced by an expansion in the Mandelstam variable t .

Ensuring that sufficient accuracy is obtained for all regions is more complicated in processes with different internal masses, or with many external particles, as there are many configurations to consider, see for example Ref. [169].

3.3. Sector decomposition

The method for evaluating Feynman integrals that is most relevant to the topics in this thesis, in particular for Chapter 4, is sector decomposition. Early works used what is now known as iterative sector decomposition (see Sec. 3.2.2) to calculate a multitude of multi-loop corrections [140–142, 170–172]. It suffers from the problem of running into infinite recursion. In Ref. [173] it was shown, based on Hironaka’s polyhedra game [174], that there exists algorithms that always terminate. The first such implementations resulted in a high number of sectors, however, and now modern programs instead use methods based on convex geometry [175–179].

3.3.1. Geometric sector decomposition

An efficient implementation of a sector decomposition algorithm based on convex geometry was presented in Ref. [179] and is implemented in PYSECDEC [12, 94, 178, 180–183]. Since it is used to evaluate all integrals in Chapter 4, we review for completeness the geometric sector decomposition algorithm, by summarizing the main equations of Chapter 7 of Ref. [179].

The target is a dimensionally regulated parameter integral

$$I = \int_{\mathbb{R}_{\geq 0}^N} \left(\prod_{i=1}^N \frac{dx_i}{x_i} x_i^{s_i} \right) f(x)^t, \quad (3.41)$$

where for simplicity we assume the integrand consists of a single polynomial $f(x)$ to be decomposed. This is for example the \mathcal{F} -polynomial in the Feynman parametrisation. We can write it on the form

$$f(x) = \sum_i^m c_i \mathbf{x}^{\mathbf{v}_i}, \quad (3.42)$$

where $\mathbf{v}_i \in \mathbb{Z}^N$ are exponent vectors and the c_i are coefficients involving the kinematic variables. The s and t are complex and contain ϵ to regulate the integrals. Feynman integrals are brought to this form by resolving the Dirac-delta constraint with, for example, the choice $\alpha(\mathbf{x}) = x_N$ (see Sec. 3.1.1). After a change of variables $z_i = \ln x_i$, the integral is

$$I = \int_{\mathbb{R}^N} d\mathbf{z} e^{\langle \mathbf{s}, \mathbf{z} \rangle} f(\mathbf{z})^t, \quad f(\mathbf{z}) = \sum_i^m c_i e^{\langle \mathbf{v}_i, \mathbf{z} \rangle}, \quad (3.43)$$

where $\langle \mathbf{a}, \mathbf{b} \rangle$ denotes the standard inner product on \mathbb{R}^N . Physical singularities are encoded in the exponent vectors \mathbf{v}_i . The Newton polytope $\mathcal{N}(f)$ of the polynomial $f(x)$ is the convex hull of the set of exponent vectors. It can be represented by the vectors \mathbf{n}_i , normal to the \mathbf{v}_i , according to

$$\mathcal{N}(f) = \bigcap_F \{ \mathbf{m} \in \mathbb{R}^N | \langle \mathbf{m}, \mathbf{n}_F \rangle + a_F \geq 0 \}, \quad (3.44)$$

where F represents an affine half-space. If $a_F = 0$, $\mathcal{N}(f)$ is called a cone. The normal fan corresponding to $\mathcal{N}(f)$ is

$$\Delta_{\mathcal{N}(f)} = \{\sigma_{\mathcal{F}} \mid \mathcal{F} \text{ is a face of } \mathcal{N}(f)\}, \quad (3.45)$$

where $\sigma_{\mathcal{F}}$ is the cone corresponding to face \mathcal{F} of $\mathcal{N}(f)$. Each cone $\sigma_{\mathcal{F}}$ is built from the normal vectors of all facets ($\dim(\mathcal{N}) - 1$ dimensional faces) that contain the face \mathcal{F} . The cones in the normal fan must be split into simplicial cones. This is known as triangulation and can be thought of as the actual decomposition step. The triangulation generates the simplicial fan $\Delta_{\mathcal{N}}^T(f)$. Its components are the simplicial cones,

$$\sigma = \left\{ \sum_{F \in \sigma} a_F \mathbf{n}_F \mid a_F \in \mathbb{R}_{\geq 0} \right\}. \quad (3.46)$$

Now the integration parameters x_i can be mapped to local facet coordinates y_F , which maps the integration domain to the unit hypercube

$$x_i = \prod_{F \in \sigma} y_F^{\langle \mathbf{n}_F, \mathbf{e}_i \rangle}. \quad (3.47)$$

Applying the change of variables to the original integral results in a sector decomposed form, which is written as a sum over all cones in the simplicial fan. After some manipulation, the decomposed integral can be expressed as [179]

$$I = \prod_{F \in \Delta_{\mathcal{N}(f)}^T} \int_0^1 \frac{dy_F}{y_F} y_F^{\langle \mathbf{n}_F, \mathbf{s} \rangle - t a_F} \left(\sum_i c_i \prod_{F \in \sigma} y_F^{\langle \mathbf{n}_F, \mathbf{v}_i \rangle + a_F} \right)^t \sum_{\substack{\sigma \in \Delta_{\mathcal{N}(f)}^T \\ \dim(\sigma) = N}} |\sigma| \prod_{F \notin \sigma} \delta(1 - y_F) \quad (3.48)$$

where $|\sigma|$ is the absolute value of the determinant of the matrix constructed from the vectors \mathbf{n}_F of σ . In the case of loop integrals, in the Feynman parametrisation, we have two polynomials (\mathcal{U} and \mathcal{F}) that must be decomposed. The extension to multiple $f_i(x)$ is achieved by considering the polytope from both polynomials $\mathcal{N}(\prod_j f_j) = \sum_j \mathcal{N}(f_j)$, known as the Minkowski sum.

3.3.2. Numerical integration

After sector decomposition, the decomposed integral can be series expanded in ϵ to reach the form in Eq. (3.35). The series coefficients are finite parameter integrals well suited for numerical integration. If there are kinematic singularities present in the integration region, a contour deformation such as in Eq. (3.26) must be performed.

In this thesis, all integrals are evaluated with quasi-Monte Carlo (QMC) integration [184], as implemented in PYSECDEC [12, 94, 178, 180–183]. We describe the method by reviewing the main equations from Ref. [181].

The parameter integrals after sector decomposition are over the unit-hypercube

$$I[f] \equiv \int_{[0,1]^d} d\mathbf{x} f(\mathbf{x}) = \int_0^1 dx_1, \dots, dx_d f(x_1, \dots, x_d), \quad (3.49)$$

where the integrand $f(\mathbf{x})$ is a finite multi-variate function on $[0, 1]^d$.

MC integration

In the classical MC approach, an estimate of $I[f]$ is computed as

$$I[f] \approx \frac{1}{n} \sum_{i=0}^{n-1} f(\mathbf{x}_i), \quad (3.50)$$

where the x_i are n randomly generated sample points. Eq. (3.50) is known as a *quadrature rule*. In the most basic version, the sample points are taken from a flat uniform distribution, $\mathbf{x}_i \in U(0, 1)^d$. Any reasonable MC integrator would instead use some form of importance sampling, which weighs the sample set according to the magnitude of $f(\mathbf{x})$. This may reduce the variance of the integrand and improve error convergence. In either case, the selection of the x_i is to some degree randomized. This allows an unbiased estimate of the variance of the integral, using the same quadrature rule

$$\sigma^2(I) = \frac{\sigma^2(f)}{n} = \frac{I(f^2) - I(f)^2}{n}. \quad (3.51)$$

The resulting error convergence of the MC estimate with respect to the number of sample points n is famously dimension independent

$$\frac{\sigma(f)}{\sqrt{n}}, \quad (3.52)$$

but scales only with the square-root of the number of sample points.

QMC integration

For integrands that fulfil certain smoothness requirements, the scaling in Eq. (3.52) is known to not be optimal. The idea of QMC integration is to choose the sample points deterministically, in a very particular way, in order to improve the error convergence. The Koksma-Hlawka inequality gives an upper bound on the integration error from the MC quadrature rule [184]

$$\left| I[f] - \frac{1}{n} \sum_{i=1}^n f(\mathbf{x}_i) \right| \leq D_n^*(P) V(f), \quad (3.53)$$

where $D_n^*(P)$ is the star discrepancy of the point-set P and $V(f)$ is the Hardy-Krause variation of f . To improve the error convergence, we can either try to minimize the discrepancy, or reduce the variance of the integrand. Selecting the sample points to be so-called *low discrepancy* sequences achieves the former and is the basis of the QMC method.

There are two main ways of constructing low-discrepancy point sets; either with *digital nets* or with *lattice rules*. Ref. [181] describes the implementation of a rank-1 shifted (R1SL) lattice rule. The integral $I[f]$ is estimated by $\bar{Q}_{n,m}[f]$, the shift-average of $Q_n^{(k)}[f]$, as follows

$$I[f] \approx \bar{Q}_{n,m}[f] \equiv \frac{1}{m} \sum_{k=0}^{m-1} Q_n^{(k)}[f], \quad Q_n^{(k)}[f] \equiv \frac{1}{n} \sum_{i=0}^{n-1} f\left(\left\{\frac{iz}{n} + \Delta_k\right\}\right), \quad (3.54)$$

where Δ_k are shift vectors, $\mathbf{z} \in \mathbb{Z}^d$ is a generating vector with elements coprime to n and the curly brackets indicate taking the fractional part of each component of the vector inside. The Δ_k are d -dimensional vectors of uniformly distributed random numbers. They reintroduce randomization to the otherwise fully deterministic quadrature rule. QMC integration based on such quadrature rules are therefore sometimes referred to as randomized QMC (RQMC). The additional evaluations due to the shifts provide an unbiased estimate of the error

$$\sigma_{n,m}^2[f] = \text{Var}(\bar{Q}_{n,m}[f]) \approx \frac{1}{m(m-1)} \sum_{k=0}^{m-1} (Q_n^{(k)}[f] - \bar{Q}_{n,m}[f])^2. \quad (3.55)$$

The extra evaluations due to the random shifts are a drawback of QMC integration. Fortunately, in practice, only 10-20 random shifts are typically necessary to obtain a reliable estimate [181]. Therefore, for sufficiently large n , the improved error scaling that we usually see for our sector decomposed Feynman integrals easily outweighs the small computational load from performing the shifts.

Using Eq. (3.53) it can be shown that QMC integration achieves an error scaling of $\mathcal{O}((\log n)^d/n)$ [184]. The scaling with respect to the number of samples is improved, but an exponential dependence on the dimension has appeared. However, if f is in certain weighted function spaces, such as a Sobolev space, much better bounds can be derived. For example, if f belongs to the space of Korobov functions, that is, periodic and α times differentiable functions, the error can be shown to converge close to $\mathcal{O}(n^{-\alpha})$ [181].

The sector decomposed integrands are generally smooth but not periodic. With the error scaling of Korobov functions in mind, a sizeable improvement can be obtained by a suitable periodizing transformation. There are many different transformations of this type, but a common one in this context is the Korobov transformation

$$\omega_{r_0, r_1}(u) = \frac{u^{r_0}(1-u)^{r_1}}{\int_0^1 dt t^{r_0}(1-t)^{r_1}} = (r_0 + r_1 + 1) \binom{r_0 + r_1}{r_0} u^{r_0}(1-u)^{r_1}, \quad (3.56)$$

usually with equal weights $r_0 = r_1$. In principle, selecting higher weights would lead to increased smoothness and better scaling by the $\mathcal{O}(n^{-\alpha})$ convergence. Too large values for the weight parameters often lead to a larger variance, however, which again increases the error. There thus exists an optimal transformation which improves the error convergence without increasing the variance too much. Experience shows that a good choice is $r_0 = r_1 = 3$, but this can vary depending on the problem.

3.3.3. pySecDec

A program that implements both sector decomposition and QMC integration to evaluate dimensionally regulated parameter integrals is PYSECDEC [12, 94, 178, 180–183]. In the later releases, PYSECDEC particularly focuses on facilitating amplitude computations. This means that evaluating loops integrals, as described in Sec. 3.3.1 and Sec. 3.3.2, is usually done in the context of an IBP system. The task is therefore not just in estimating one I , but in computing a linear combination of master integrals

$$\mathcal{A} = \sum_i c_i(\mathbf{s}, D) I_i, \quad (3.57)$$

where the IBP coefficients $c_i(\mathbf{s}, D)$ are functions of the kinematics and the dimension. After sector decomposition, each master integral at each order in ϵ is in turn a linear combination of *sector integrals*. To avoid introducing too much notation, we overload the above definition with $\mathcal{A} = \sum_i c_i I_i$, where the c_i are now a combination of IBP coefficients and integral prefactors, and the I_i are now sector integrals.

To optimize such computations, PYSECDEC uses *weighted evaluation*. Since the values of both the coefficients and the integrals vary for each kinematic point, a data-driven importance sampling is performed before each run, in order to be able to allocate an appropriate amount of computing resources to each integral. The integrals that have a good combination of contributing more to the amplitude while being fast to evaluate, are assigned larger QMC lattices. The optimization problem is formulated with Lagrange multipliers [12]. The total integration time

$$T \equiv \sum_i \tau_i n_i, \quad (3.58)$$

where τ_i is the time it takes to evaluate the integrand of I_i once, is minimized subject to a precision constraint on the amplitude

$$\text{var}(\mathcal{A}) = \sum_i |c_i|^2 \frac{w_i}{n_i^\alpha} \stackrel{!}{=} V, \quad (3.59)$$

where α is the scaling parameter from Sec. 3.3.2 and w_i is a convergence constant. Combining the two constraints, the minimization problem is specified by

$$L \equiv T + \sum_k \lambda (\text{var}(\mathcal{A}) - V), \quad \text{and} \quad \frac{\partial L}{\partial \{n_i, \lambda_k\}} = 0. \quad (3.60)$$

The solution for the number of sample points that should be assigned to each integral is [12]

$$\begin{aligned} \lambda &= \frac{1}{\alpha} \left(\frac{1}{V} \sum_k (|c_k|^2 w_k \tau_k^\alpha)^{\frac{1}{\alpha+1}} \right)^{\frac{\alpha+1}{\alpha}}, \\ n_i &= \left(\frac{\alpha w_i}{\tau_i} \lambda |c_i|^2 \right)^{\frac{1}{\alpha+1}}. \end{aligned} \quad (3.61)$$

A conservative choice for the scaling parameter is $\alpha = 2$ (see the discussion at the end of Sec. 3.3.2). The evaluation speeds τ_i and convergence constants w_i are initially estimated by evaluating each integral on a small set of presamples, and then continuously updated as the lattice sizes are increased.

In Ref. [12] we present the latest version of PYSECDEC. The new version introduces a new QMC integrator **Disteval**, new median lattice rules, improvements to the expansion by regions module as well as improvements to the interface of defining amplitudes as sums of integrals. We list and briefly describe the main new features here.

- The **Disteval** integrator brings speed-up by up to an order of magnitude (both on the CPU and GPU side) through various code improvements and extended support for advanced compiler options. Additionally, the new integrator allows defining IBP coefficients that are computed to arbitrary precision with GINAC. Symbolic coefficients are saved to file and only expanded in ϵ at evaluation.

- Median lattice rules are introduced to cure the problem of *unlucky lattices*. When continuously estimating the τ_i and w_i parameters as described above, it can happen that a larger lattice randomly performs very poorly, which spoils the data-driven estimate of the convergence parameters. This is not surprising, since the $\mathcal{O}(n^{-\alpha})$ scaling law is only asymptotic. Locally, increasing a lattice size does not necessarily lead to an improvement. It was shown in Ref. [185] that median lattice rules avoids this problem, to an extent. The idea is to instead of using one $Q_n^{(k)}$ for the integral estimate, based on one generating vector \mathbf{z} , we take instead the median

$$M_{n,r}(f) = \text{median}(Q_{n,\mathbf{z}_1}(f), \dots, Q_{n,\mathbf{z}_r}(f)). \quad (3.62)$$

This avoids selecting outliers at the expense of more integrand evaluations. In practice, we see that examples that are known to particularly suffer from unlucky lattices are greatly improved with this construction.

- The improvements to expansion by regions is in terms of automatically inserting extra regulators to integrals that need them after expansion. The extra regulators are shifts to the propagator exponents by a small parameter, $\nu_1 \rightarrow \nu_1 + \delta_1, \dots, \nu_N \rightarrow \nu_N + \delta_N$. In parameter space, this is achieved by multiplying the integrand by $x_1^{\delta_1} \cdots x_N^{\delta_N}$.

3.4. Flow-time integrals with ftint

In this section we present FTINT [13], a program to calculate flow-time integrals that appear within the gradient-flow formalism (GFF) [186–190]. The GFF was originally introduced in QCD to regulate UV divergences on the lattice. One of the main advantages is that composite operators do not require UV renormalisation and thus do not mix under renormalisation group-running [191]. This simplifies the comparison of results from lattice QCD and perturbative calculations.

One application of the GFF that involves perturbative calculations is the short-flow-time expansion (SFTX), where composite operators of flowed fields are expressed in terms of regular operators via matching coefficients which can be determined perturbatively. The integrals that appear in the SFTX, so-called flow-time integrals, are very similar to the Feynman loop integrals discussed in this chapter. The only two differences are

1. An exponential factor in the integrand that depends on the loop momenta, masses and *flow-time variables*.
2. Additional integrations over these flow-time variables.

Despite these very small differences, the wide-spread selection of tools that we take for granted in the context of evaluating multi-loop Feynman integrals, had prior to FTINT not been made applicable to flow-time integrals¹⁹. With FTINT, we provide an interface to PYSECDEC that maps (massive) flow-time integrals with vanishing external momenta²⁰ to a form that PYSECDEC can handle, similar to Eq. (3.41). In its current state FTINT is limited to one-, two- and three-loop integrals, see Fig. 3.2.

¹⁹Specifically the evaluation part. Amplitude generation and IBP reduction had already been brought to a mature state in the context of flow-time integrals.

²⁰This is a phenomenologically relevant class of integrals. For example, they enter into the action density $\langle G_{\mu\nu} G_{\mu\nu} \rangle$.

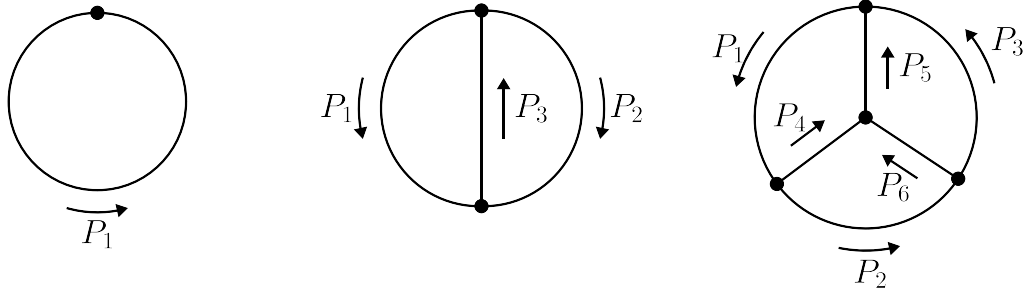


Figure 3.2.: One-, two-, and three-loop topologies of the integrals considered in this section.

Following the notation of Ref. [13], the flow-time integrals we target can be represented by [192, 193]

$$f(\mathbf{c}, \mathbf{a}, \mathbf{b}) = (4\pi t)^{LD/2} t^{-b} \int_{[0,1]^f} d\mathbf{u} \mathbf{u}^{\mathbf{c}} \int_{\mathbf{p}} \frac{\exp[-t \sum_{i=1}^k a_i P_i^2]}{(P_1^2 + m_1^2)^{b_1} \cdots (P_k^2 + m_k^2)^{b_k}}, \quad (3.63)$$

where $\mathbf{p} = \{p_1, \dots, p_L\}$ is the set of loop momenta, $\mathbf{b} = \{\{b_1, m_1\}, \dots, \{b_k, m_k\}\}$ collects the indices $b_i \in \mathbb{Z}$ and masses $m_i \in \mathbb{R}$. The prefactor t^{-b} , with $b = \sum_{i=1}^k b_i$, in Eq. (3.63) thus compensates the mass dimension of the integral, such that $f(\mathbf{c}, \mathbf{a}, \mathbf{b})$ is dimensionless. For massless propagators we use the special notation $\{b_i, 0\} \rightarrow b_i$. The k functions $a_i \in \mathbf{a}$ are real-valued polynomials of the (dimensionless) flow-time variables $\mathbf{u} = \{u_1, \dots, u_f\}$ that are non-negative on the hypercube $[0, 1]^f$. Furthermore, we define

$$\mathbf{u}^{\mathbf{c}} := u_1^{c_1} \cdots u_f^{c_f}, \quad (3.64)$$

where $\mathbf{c} = \{c_1, \dots, c_f\}$ is a set of non-negative integers. The integration measure over the loop momenta is

$$\int_{\mathbf{p}} := \int_{p_1} \cdots \int_{p_L}, \quad \text{with} \quad \int_p = \int \frac{d^D p}{(2\pi)^D}. \quad (3.65)$$

The P_i in Eq. (3.63) are linear combinations of the loop momenta. It implies the very convenient feature that the number of propagators k in Eq. (3.63) is completely determined by the number of loops L as

$$k = \begin{cases} 1 & \text{for } L = 1, \\ 3(L - 1) & \text{for } L \geq 2. \end{cases} \quad (3.66)$$

3.4.1. Calculation method

The integral in Eq. (3.63) must be brought to a form that PYSECDEC is able to parse. We thus need to introduce a suitable parametrisation and integrate over the loop momenta. For this purpose, we use Schwinger parameters. The result is an integral over both flow-time variables $u \in (0, 1)$ and Schwinger parameters $x \in (0, \infty)$. Either must be mapped to the other interval as the sector decomposition algorithms expect all integration variables to be in the same interval. The procedure is best illustrated with a few examples.

Massless one-loop

A general one-loop flow-time integral with massless internal lines is

$$f(\mathbf{c}, \{a\}, \{b\}) = (4\pi t)^{D/2} t^{-b} \int_{[0,1]^f} d\mathbf{u} \mathbf{u}^{\mathbf{c}} \int_p \frac{\exp[-tap^2]}{(p^2)^b}. \quad (3.67)$$

If $b = 0$ the momentum integration can be performed and the expression can be passed to PYSECDEC directly. If $b > 0$ we introduce Schwinger parameters and obtain

$$\begin{aligned} f(\mathbf{c}, \{a\}, \{b\}) &= \frac{(4\pi t)^{D/2}}{(b-1)!} \int_0^\infty dx x^{b-1} \int_{[0,1]^f} d\mathbf{u} \mathbf{u}^{\mathbf{c}} \int_p \exp\{-t[a+x]p^2\} = \\ &= \frac{1}{(b-1)!} \int_0^\infty dx x^{b-1} \int_{[0,1]^f} d\mathbf{u} \mathbf{u}^{\mathbf{c}} [a+x]^{-D/2}, \end{aligned} \quad (3.68)$$

where in the last step we performed Gaussian integration over the loop momentum. Now we remap the Schwinger parameters²¹ to the unit-hypercube by splitting $(0, \infty) = (0, 1) \cup (1, \infty)$ and applying $x \rightarrow 1/x$ in the second interval. The result is an integral that can be passed to PYSECDEC

$$\begin{aligned} f(\mathbf{c}, \{a\}, \{b\}) &= \frac{1}{(b-1)!} \int_{[0,1]^f} d\mathbf{u} \mathbf{u}^{\mathbf{c}} \\ &\times \int_0^1 dx (x^{b-1} [a+x]^{-D/2} + x^{D/2-1-b} [xa+1]^{-D/2}). \end{aligned} \quad (3.69)$$

If $b < 0$ we use the identity

$$(p^2)^{-b} = (-t)^b \frac{\partial^{-b}}{\partial x^{-b}} e^{-xtp^2} \Big|_{x=0} \quad (3.70)$$

and obtain

$$\begin{aligned} f(\mathbf{c}, \{a\}, \{b\}) &= (-1)^b \frac{\partial^{-b}}{\partial x^{-b}} \int_{[0,1]^f} d\mathbf{u} \mathbf{u}^{\mathbf{c}} [a+x]^{-D/2} \Big|_{x=0} \\ &= P(-b) (-1)^b \int_{[0,1]^f} d\mathbf{u} \mathbf{u}^{\mathbf{c}} [a]^{-D/2+b}, \end{aligned} \quad (3.71)$$

where

$$P(m) = \prod_{k=1}^m (1 - k - D/2) = \frac{\Gamma(1 - D/2)}{\Gamma(1 - D/2 - m)}. \quad (3.72)$$

This expression can again be directly passed to PYSECDEC.

Massless multi-loop

To understand how to integrate out the loop momenta after Schwinger parametrisation in the multi-loop case, some auxiliary notation is helpful. We define a symmetric square matrix $A(\mathbf{a})$ by

$$\mathbf{p}^T A(\mathbf{a}) \mathbf{p} = \sum_{i=1}^k a_i P_i^2. \quad (3.73)$$

It is determined by the combinations of loop momentum appearing in the propagators. Since the number of propagators at each loop order is exactly known as in Eq. (3.66), we can define it explicitly at

²¹We could equally well have chosen to map the flow-time variables to $[0, \infty]^f$.

one loop:

$$\begin{aligned} P_1 &= p_1, \\ \Rightarrow A(\mathbf{a}) &= a_1, \end{aligned} \quad (3.74a)$$

two loops:

$$P_1 = p_1, \quad P_2 = p_2, \quad P_3 = p_1 + p_2, \quad (3.75a)$$

$$\Rightarrow A(\mathbf{a}) = \begin{pmatrix} a_1 + a_3 & a_3 \\ a_3 & a_2 + a_3 \end{pmatrix} \quad (3.75b)$$

and three loops:

$$\begin{aligned} P_1 &= p_1, & P_2 &= p_2, & P_3 &= p_3, \\ P_4 &= p_1 - p_2, & P_5 &= p_1 - p_3, & P_6 &= p_2 - p_3, \end{aligned} \quad (3.76a)$$

$$\Rightarrow A(\mathbf{a}) = \begin{pmatrix} a_1 + a_4 + a_5 & -a_4 & -a_5 \\ -a_4 & a_2 + a_4 + a_6 & -a_6 \\ -a_5 & -a_6 & a_3 + a_5 + a_6 \end{pmatrix}. \quad (3.76b)$$

Finally, we define a function

$$F(\mathbf{a}, \mathbf{x}) = (4\pi t)^{LD/2} \int_{\mathbf{p}} \exp \left\{ -t \sum_{i=1}^k [a_i + x_i] P_i^2 \right\}. \quad (3.77)$$

Using Eq. (3.73) we identify it as Gaussian over the loop momenta which leads to

$$F(\mathbf{a}, \mathbf{x}) = [\det A(\mathbf{a} + \mathbf{x})]^{-D/2}. \quad (3.78)$$

To obtain a compact master formula we introduce a notation for the b -indices as follows

$$\begin{aligned} b_i &> 0 & \text{for } i \in I_{\text{int}}, \\ b_i &< 0 & \text{for } i \in I_{\text{diff}}, \\ b_i &= 0 & \text{for } i \in I_0. \end{aligned} \quad (3.79)$$

For the vanishing indices $b_i = 0$, we can simply set the corresponding Schwinger parameters to zero in Eq. (3.78), $x_i = 0$. For the negative indices $b_i < 0$, on the other hand, we use Eq. (3.70), meaning that we need to take the derivative w.r.t. $-x_i$ at $x_i = 0$. Note that n derivatives acting on Eq. (3.78) produce n terms of the form

$$[\det A(\mathbf{a} + \mathbf{x})]^{-D/2-k} g_k(\mathbf{a}, \mathbf{x}, D), \quad k \in \{1, \dots, n\}, \quad (3.80)$$

where g_k is polynomial in its arguments at most of order x_i^k for each x_i . Finally, for the positive indices $b_i > 0$, we integrate over x_i and multiply by $1/(b_i - 1)!$. Combining everything we obtain a master formula for a flow-time integral, taken over Schwinger parameters and flow-time variables

$$\begin{aligned} f(\mathbf{c}, \mathbf{a}, \mathbf{b}) &= \left[\prod_{j \in I_{\text{int}}} \frac{1}{(b_j - 1)!} \int_0^\infty dx_j x_j^{b_j-1} \right] \times \\ &\times \int_{[0,1]^f} d\mathbf{u} \mathbf{u}^{\mathbf{c}} \left[\left(\prod_{i \in I_{\text{diff}}} \frac{\partial^{-b_i}}{\partial x_i^{-b_i}} \right) [\det A(\mathbf{a} + \mathbf{x})]^{-D/2} \right]_{x_k=0 \text{ for } k \in (I_{\text{diff}} \cup I_0)}. \end{aligned} \quad (3.81)$$

To be able to pass this to PYSECDEC, we split the integration region for the Schwinger parameters into $x_j \in [0, 1] \cup (1, \infty]$. The integrand in Eq. (3.81) thus consists of polynomials of the x_i with $i \in I_{\text{int}}$ and the \mathbf{u} , raised to non-integer powers which can be passed to PYSECDEC for integration.

Massive propagators

With the massless case already defined, including masses is straightforward. The only modification to the master formula above is a factor

$$\exp \left(-t \sum_{j \in I_{\text{int}}} x_j m_j^2 \right). \quad (3.82)$$

Masses corresponding to propagators with negative indices do not need to be considered as they can be reduced to other known integrals via the binomial formula

$$(p^2 + m^2)^{-b} = \sum_{n=0}^{-b} \binom{-b}{n} (p^2)^n (m^2)^{-b-n} \quad \text{for } b \leq 0. \quad (3.83)$$

A technical issue with including massive propagators is that after remapping $x_j \rightarrow 1/x_j$ in the second interval of the split, factors of $e^{-tm_j^2/x_j}$ appear. This expression is clearly finite in the limit $x_j \rightarrow 0$, but it still creates some practical issues in PYSECDEC, for example when defining subtraction terms at $x_j = 0$. To avoid all of this, a regulator is introduced that can be set to a small number at integration. We shift each exponential factor by

$$e^{-tm_j^2/x_j} \rightarrow e^{-tm_j^2/(x_j+\delta)}. \quad (3.84)$$

The effect of including massive propagators is a scaling of the integrand by a finite, positive function. It therefore does not interfere with sector decomposition and does not generate any additional sectors. This means that the extra regulator in Eq. (3.84) has virtually no impact on performance. By default, $\delta = 10^{-10}$ at integration but can be varied by the user to ensure that the result is independent of its value²².

3.4.2. Using ftint

Following the spirit of PYSECDEC, the evaluation of an integral with FTINT is divided into two steps. In the first step, the integral is sector decomposed and a C++ integration library is generated and compiled. The integration library can then be evaluated many times for different values of the internal masses, or at different precision targets, for example.

The program can be downloaded from gitlab.com/ftint where installation instructions can also be found. FTINT is run via a command-line interface. For example, to generate the integration library the following command is called

```
$ python3 <ftint_path>/ftint_pySecDec.py \
    <ft_integrals> [<options>]
```

Listing 3.1: Command for sector decomposition.

²²The inclusion of δ will always generate a small systematic error, but if it is small enough it can be considered negligible compared to the statistical error of the MC integration. The integral is in this way practically ‘independent’ of the value of δ .

The `<ftint_path>` is the path to the source code and `<ft_integrals>` is the input integral, following the convention in Eq. (3.63). It can also be the path to a file containing a list of integrals, defined with the same notation. In this way, FTINT can for example evaluate sums of integrals appearing in an IBP system. Including masses is done by an extra index next to each propagator power. For example, the input to the following massive two-loop integral is

$$\begin{aligned} & f[\{0,1\},\{u_2-u_1*u_2,2,u_2\},\{-1,\{2,2\},\{1,3\}\}] = \\ & = f(\{0,1\},\{u_2-u_1*u_2,2,u_2\},\{-1,\{2,m_2\},\{1,m_3\}\}) = \\ & (4\pi t)D \int_{p,k} \int_{u_{1,2}} u_2 \frac{p^2}{(k^2+m_2^2)^2((p+k)^2+m_3^2)} e^{-t[(-u_1u_2+u_2)p^2+2k^2+u_2(p+k)^2]}. \end{aligned} \quad (3.85)$$

We can generate and compile the integration libraries for this integral with the command

```
$ python3 <ftint_path>/ftint_pySecDec.py \
    'f[{0,1},{u2-u1*u2,2,u2},{-1,{2,2},{1,3}}]'
```

Listing 3.2: Sector decomposition of the integral in Eq. (3.85).

and then integrate it with the command

```
$ python3 <ftint_path>/ftint_integrate.py \
    ftint_out_0 --masses=0,2.5,3
```

Listing 3.3: Numerical evaluation of the integral in Eq. (3.85).

Where the values of the masses were chosen to be $m_2^2 = 2.5/t$ and $m_3^2 = 3/t$. The output of the integration is available in multiple formats and is stored in a subdirectory of `ftint_out_0`. For example, the Mathematica output format is

```
1 (*
2 produced by ftint, version 1.0, Fri May 24 13:50:34 2024
3 *)
4 {
5 (* integral 1 [ m2**2 = 2.5/t, m3**2 = 3/t ]: *)
6 f[{0,1},{u2-u1*u2,2,u2},{-1,{2,2},{1,3}}] -> (
7   +eps^-2*(+0.0000000000000000*10^+00+0.0000000000000000*10^+00*I)
8   +eps^-2*(+0.0000000000000000*10^+00+0.0000000000000000*10^+00*I)*plusminus
9   +eps^-1*(+1.1266529421611552*10^-02+0.0000000000000000*10^+00*I)
10  +eps^-1*(+2.3027953579787037*10^-10+0.0000000000000000*10^+00*I)*plusminus
11  +eps^0*(+1.7797179680747620*10^-04+0.0000000000000000*10^+00*I)
12  +eps^0*(+1.0762786605309655*10^-09+0.0000000000000000*10^+00*I)*plusminus
13  +eps^1*(+6.8615123921316060*10^-03+0.0000000000000000*10^+00*I)
14  +eps^1*(+1.1513874186746525*10^-09+0.0000000000000000*10^+00*I)*plusminus
15  +eps^2*(-5.5497372202654837*10^-03+0.0000000000000000*10^+00*I)
16  +eps^2*(+7.0664076906102803*10^-09+0.0000000000000000*10^+00*I)*plusminus
17  +eps^3*(+5.2760242194534335*10^-03+0.0000000000000000*10^+00*I)
18  +eps^3*(+3.2253823531270829*10^-08+0.0000000000000000*10^+00*I)*plusminus
19 )
20 }
```

Listing 3.4: Output file of the numerical integration.

In addition to specifying the values of masses, FTINT can access all decomposition and integration options within PYSECDEC. It is also possible to pass any C++ (CUDA) compiler flags to the generation by specifying the `--CXX_flags` (`--CUDA_flags`) option. For a full list of options we refer to the documentation within Ref. [13].

IBP example

To demonstrate the intended input format of FTINT we verify a simple IBP relation as an example. The input is a file, `ibp_rule.in`, containing a `Mathematica` replacement rule.

```
1 {f[{},{0,0,0,1,1,1},{-1,1,1,1,1,1}] ->
2 -f[{},{0,0,0,1,1,1},{0,1,1,1,1,0}]/2
3 -(f[{},{0,0,0,1,1,1},{1,0,1,1,0,0}]/(1-n/4)
4 -(f[{},{0,0,0,1,1,1},{1,1,0,0,0,0}]/(2*(1+(-7/12+n/12)*n)))}
```

Listing 3.5: A three-loop IBP identity in `Mathematica` format.

The file path can be passed to `ftint_pySecDec.py`, which will generate and compile individual integration libraries for each integral in the input IBP relation. We generate the integration libraries up to ϵ^1 with the command

```
$ python3 <ftint_path>/ftint_pySecDec.py \
    ibp_rule.in --epsorder=1
```

The output is by default placed in `ftint_out_0` where we find the four compiled integration libraries, as well as other information required by FTINT for the integration.

```
\-- ftint_out_0
  |-- integral_information.json
  \-- secdec
    |-- secdec_ft_integral_1
    |-- secdec_ft_integral_2
    |-- secdec_ft_integral_3
    \-- secdec_ft_integral_4
```

The integration can now be performed with the command

```
$ python3 <ftint_path>/ftint_integrate.py ftint_out_0
```

The output is stored in `ftint_out_0/results_0`. It contains the `Mathematica` file `mathf_out.m` where the result for each integral is defined as a replacement rule.

```
1 (*
2   produced by ftint, version 1.0, Fri May 24 09:21:05 2024
3   *)
4   {
5   (* integral 1 : *)
6   f[{},{0,0,0,1,1,1},{-1,1,1,1,1,1}] -> (
7     +eps^-1*(+1.4384102482242656*10^-01+0.0000000000000000*10^+00*I)
8     +eps^-1*(+7.5901486511209429*10^-09+0.0000000000000000*10^+00*I)*plusminus
9     +eps^0*(+9.1030039859574119*10^-01+0.0000000000000000*10^+00*I)
10    +eps^0*(+7.7645751796859680*10^-06+0.0000000000000000*10^+00*I)*plusminus
11    +eps^1*(+3.6458111371273918*10^+00+0.0000000000000000*10^+00*I)
12    +eps^1*(+4.2456809764811359*10^-05+0.0000000000000000*10^+00*I)*plusminus
13  ),
14  (* integral 2 : *)
15  f[{},{0,0,0,1,1,1},{0,1,1,1,1,0}] -> (
16    +eps^-1*(+2.8768207244049038*10^-01+0.0000000000000000*10^+00*I)
17    +eps^-1*(+1.2597382329360634*10^-11+0.0000000000000000*10^+00*I)*plusminus
18    +eps^0*(+1.4717502294126590*10^+00+0.0000000000000000*10^+00*I)
19    .
20    .
21    .
```

Listing 3.6: Result for the integrals of 3.5. Only the first few lines are shown.

The IBP relation can now be verified by inserting the list of replacement rules in the output back into the original IBP relation in **Mathematica**. This output format makes the integration of FTINT into existing workflows straightforward. In addition, the integration results are also available in the JSON and YAML formats. It is also possible to define customized output formats, see Appendix A of Ref. [13].

CHAPTER 4

Positive Feynman Integrands

In this chapter we present a new idea of resolving integrable singularities of Feynman loop integrals [14]. In Sec. 3.1.2 the classic treatment with contour deformation (CD) is outlined. Its advantage is that it is algorithmic and can be made to work in many cases. Its downside is that it is computationally expensive and leads to long integration times and often quite significant loss of precision.

As a motivation we thus consider Fig. 4.1, which shows the numerical precision obtained from integrating the one-loop massless pentagon integral with 10^5 QMC points with PYSECDEC, for values of the kinematic invariant $s_{12} \in (-10, 10)$, with the other invariants fixed to negative values; $(s_{23}, s_{34}, s_{45}, s_{51}) = (-4, -2, -6, -3)$. The details of this integral are given in Sec. 4.3.1. For now, all we need to know is that the integral has a so-called Euclidean region when $s_{12} < 0$ and a so-called Minkowski region when $s_{12} > 0$. In the Minkowski regime integrable singularities appear and must be resolved, in this case by using contour deformation. The result is a huge precision drop for all kinematic configurations where $s_{12} > 0$. This can be explained by the fact that the deformed integrand changes sign across the parameter space in this region. This leads to cancellations between positive and negative contributions to the integral and the precision drops. Moreover, the deformed integrand is more complicated and takes longer to evaluate. For this example, one integrand evaluation is on average $4.54/1.7 \approx 2.67x$ slower when evaluating with contour deformation²³. Contour deformation thus produces slower evaluations that yield less accurate results. The new method presented in this chapter rewrites Feynman integrals into a sum of integrals over positive integrands, multiplied by complex prefactors. The performance degradation seen in the Minkowski regime is significantly alleviated and the evaluation time scales much better with respect to the number of internal lines, compared to when using contour deformation.

²³This slowdown becomes significantly worse for multi-loop integrals that have many internal lines N . The contour deformation includes a Jacobian that involves partial derivatives of \mathcal{F} with respect to each Feynman parameter. This results in an $(N-1) \times (N-1)$ determinant that has to be evaluated at each QMC point.

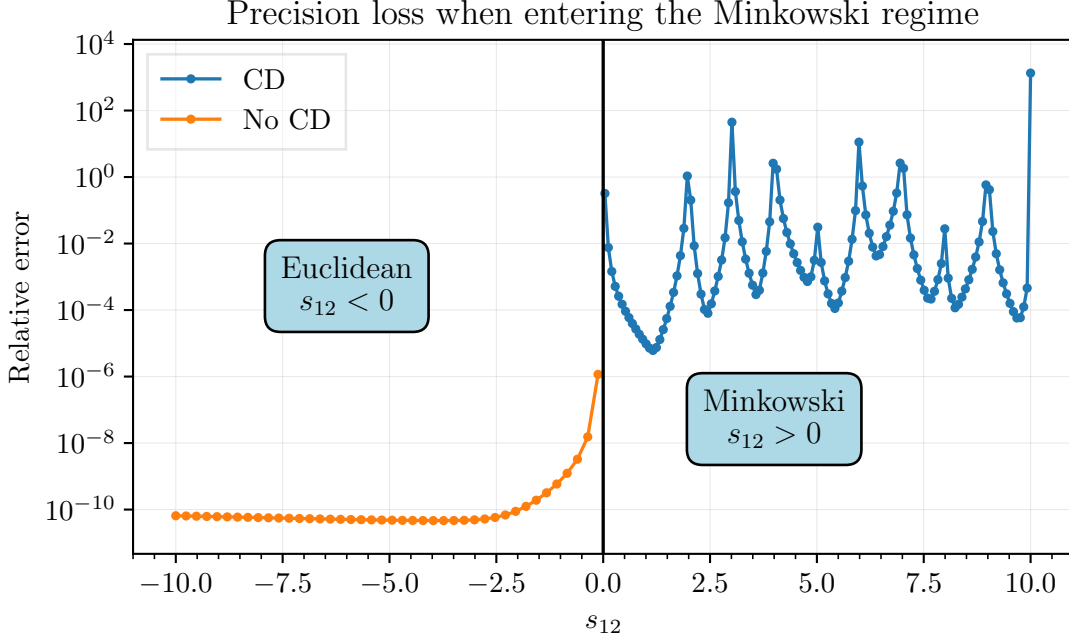


Figure 4.1.: Numerical precision obtained from 10^5 QMC points for the real part of the fully on-shell, one-loop pentagon with massless internal lines. The kinematics are specified by $s_{12} \in (-10, 10)$ and $(s_{23}, s_{34}, s_{45}, s_{51}) = (-4, -2, -6, -3)$. The integral enters the Minkowski regime when $s_{12} > 0$, leading to significant loss of precision. For $s_{12} < 0$, the integral can be evaluated without contour deformation leading to faster integrand evaluations. Evaluating with contour deformation is on average about $4.54/1.7 \approx 2.67x$ times slower for this example.

The outline of this chapter is as follows. We start with some preliminary definitions in Sec. 4.1. Then we define the general method in Sec. 4.2, where an algorithm is presented that can be applied to a certain class of Feynman integrals. In Sec. 4.3.1 and Sec. 4.3.2 we apply the method to resolve integrable singularities of several massless and massive integrals. Finally, we benchmark the examples in Sec. 4.4 and give an outlook in Sec. 4.5.

4.1. Preliminaries

A thorough review of parametric representations and divergences of Feynman integrals is given in Sec. 3.1. In this section, all examples happen to feature integrals without dots, pinches or numerators, i.e. we have $\nu_i = 1, \forall i$. For brevity, we therefore make use of the notation $I(1, \dots, 1; \mathbf{s}) := J(\mathbf{s})$ in this chapter. In the present case the Feynman parametrisation simplifies slightly

$$J(\mathbf{s}) = (-1)^\nu \Gamma(\nu - LD/2) \lim_{\delta \rightarrow 0^+} \int_{\mathbb{R}_{\geq 0}^N} \prod_{i=1}^N dx_i \frac{\mathcal{U}(\mathbf{x})^{\nu - (L+1)D/2}}{(\mathcal{F}(\mathbf{x}; \mathbf{s}) - i\delta)^{\nu - LD/2}} \delta(1 - \alpha(\mathbf{x})), \quad (4.1)$$

where ν is in this case equal to the number of internal lines. The remaining notation, in particular the $\mathcal{U}(\mathbf{x})$ and $\mathcal{F}(\mathbf{x}; \mathbf{s})$ polynomials, is defined in Sec. 3.1. An important

property is that $\mathcal{F}(\mathbf{x}; \mathbf{s})$ contain terms with quadratic Feynman parameters if the integral has massive propagators, while massless integrals have $\mathcal{F}(\mathbf{x}; \mathbf{s})$ polynomials that are at most linear in each parameter. Furthermore, we note that $\mathcal{U}(\mathbf{x})$ consists purely of positive monomials of Feynman parameters with positive coefficients and is therefore manifestly positive in $\mathbb{R}_{>0}^N$. The $\mathcal{F}(\mathbf{x}; \mathbf{s})$ polynomial is built from monomials with both positive and negative sign depending on the value of the invariants \mathbf{s} .

Both $\mathcal{U}(\mathbf{x})$ and $\mathcal{F}(\mathbf{x}; \mathbf{s})$ may vanish on the intersection of the hyperplane defined by the Dirac-delta function with the coordinate hyperplanes, or, equivalently, for subsets of \mathbf{x} vanishing. Depending on the propagator powers, ν , and the space-time dimension, D , this can give rise to UV and IR divergences. When the invariants \mathbf{s} are chosen such that $\mathcal{F}(\mathbf{x}; \mathbf{s})$ contains monomials with different signs, the $\mathcal{F}(\mathbf{x}; \mathbf{s})$ polynomial can additionally vanish and/or change sign on hypersurfaces within the integration domain, $\mathbb{R}_{>0}^N$. The vanishing of $\mathcal{F}(\mathbf{x}; \mathbf{s})$ within the integration domain is usually associated with kinematic (pseudo-)thresholds.

We distinguish between a *manifestly same-sign kinematic regime*, where all monomials in $\mathcal{F}(\mathbf{x}; \mathbf{s})$ share the same sign and a more general *same-sign regime*²⁴, where $\mathcal{F}(\mathbf{x}; \mathbf{s})$ retains a definite sign throughout the integration domain, $\mathbb{R}_{>0}^N$, despite individual monomials having mixed signs. In the most general situation $\mathcal{F}(\mathbf{x}; \mathbf{s})$ vanishes on hypersurfaces in $\mathbb{R}_{>0}^N$ and has no fixed definite sign in the domain of integration, we will refer to this as a *mixed-sign (or Minkowski) regime*.

4.2. Method

In this section, we propose a procedure for evaluating Feynman parametrised integrals in the mixed-sign (Minkowski) regime, i.e. where $\mathcal{F}(\mathbf{x}; \mathbf{s})$ has no definite sign for $\mathbf{x} \in \mathbb{R}_{>0}^N$, without deforming the integration contour into the complex-plane. We introduce the general idea in Sec. 4.2.1 and present a constructive algorithm valid for a specific class of Feynman integrals in Sec. 4.2.2. We show that when this form can be achieved, the resulting integrands are strictly non-negative and the analytic continuation of the resulting integrals becomes trivial. In Sec. 4.2.3 we briefly discuss the application of the concept to general Feynman integrals.

4.2.1. Overview

The central idea is to construct transformations of the Feynman parameters such that the variety of the \mathcal{F} -polynomial (the hypersurface defined by $\mathcal{F}(\mathbf{x}; \mathbf{s}) = 0$) is mapped to the boundary of the integration domain. After this, \mathcal{F} only vanishes on the integration boundary. Any singularities resulting from this can be algorithmically dealt with using existing methods, such as sector decomposition.

The new procedure involves splitting the integration domain into regions where $\mathcal{F} > 0$ and $\mathcal{F} < 0$ and integrating these regions separately. We may also further subdivide the positive and negative regions for technical or computational ease (see the massive examples in Sec. 4.3.2). For the regions where \mathcal{F} is negative, we factor out a minus sign from \mathcal{F} , ensuring the Feynman $i\delta$ -prescription is respected; after this, we will have only

²⁴The terms *Euclidean* and *pseudo-Euclidean* are sometimes used in the literature to refer to this regime.

non-negative integrands. The resulting general decomposition of the original Feynman integral, $I(\mathbf{s})$, is

$$J(\mathbf{s}) = \sum_{n_+=1}^{N_+} J^{+,n_+}(\mathbf{s}) + \lim_{\delta \rightarrow 0^+} (-1 - i\delta)^{-(\nu - LD/2)} \sum_{n_-=1}^{N_-} J^{-,n_-}(\mathbf{s}), \quad (4.2)$$

where we have allowed for both the positive and negative regions to be subdivided into N_+ and N_- sub-regions respectively. After this decomposition, the imaginary part of the original integral $J(\mathbf{s})$ is fully determined by the contribution from the negative region(s) in Eq. (4.2) and, furthermore, is a result of the $\lim_{\delta \rightarrow 0^+} (-1 - i\delta)^{-(\nu - LD/2)} = \exp[i\pi(\nu - LD/2)]$ factor multiplying the purely real $J^-(\mathbf{s})$ contribution(s). We remark that, in many cases, the division can be chosen such that there is only a single positive and a single negative region (see the algorithm in Sec. 4.2.2 and the massless examples in Sec. 4.3.1).

It has long been known that for integrals in a kinematic regime where \mathcal{F} is strictly non-positive in the integration domain, a minus sign can be factored out and the resulting non-negative (Euclidean) integrand can be evaluated. As a trivial example, consider the massless bubble in the physical $p^2 = s > 0$ regime,

$$\begin{aligned} J_{\text{bub}, m=0}(\mathbf{s}) &= \lim_{\delta \rightarrow 0^+} \Gamma(\epsilon) \int_{\mathbb{R}_{\geq 0}^2} dx_1 dx_2 \frac{(x_1 + x_2)^{-2+2\epsilon}}{(-sx_1x_2 - i\delta)^\epsilon} \delta(1 - \alpha(\mathbf{x})) \\ &= \lim_{\delta \rightarrow 0^+} (-1 - i\delta)^{-\epsilon} \Gamma(\epsilon) \int_{\mathbb{R}_{\geq 0}^2} dx_1 dx_2 \frac{(x_1 + x_2)^{-2+2\epsilon}}{(sx_1x_2)^\epsilon} \delta(1 - \alpha(\mathbf{x})) \\ &= \lim_{\delta \rightarrow 0^+} (-1 - i\delta)^{-\epsilon} J_{\text{bub}, m=0}^-(\mathbf{s}), \end{aligned} \quad (4.3)$$

where, in the final line we have translated this simple manipulation into the language of our decomposition given in Eq. (4.2). The procedure essentially generalises this idea by mapping a generic integral into integrals which are already positive (where $\mathcal{F} > 0$ originally) plus cases like this where $\mathcal{F} < 0$ and we can factor out the prescription to generate positive integrands.

In order to construct the integrands appearing in Eq. (4.2), we must ensure that our transformations do not spoil the non-negativity of \mathcal{U} (nor introduce zeroes of \mathcal{U} within the integration domain) as well as avoiding transformations with Jacobian determinants which break the positivity of the resulting integrand within (but not necessarily on the boundary of) the integration domain. Furthermore, we must check that the transformations applied do not miss any regions from the original integration domain as well as prohibiting transformations that map regions from outside the original integration domain into the new domain.

These demands may initially appear quite constraining and hence, one might assume that the transformations are potentially difficult to construct. However, we will show that for a large number of massless integrals (a sample of which are detailed in Sec. 4.3.1), the procedure is algorithmic. Additionally, geometric visualisations can prove extremely useful for building an understanding of how the transformation should appear, which aids its construction. We adopt this approach on a case-by-case basis for the resolution of the massive integrals in Sec. 4.3.2. In its current form, this approach

suffers from the disadvantage that it is difficult to visualise beyond four propagators, although this does not present an in-principle obstacle to the concept.

The interplay between the positive and negative contributions in our dissection also allows us to obtain an understanding of the structure of the original integral from a new perspective. For example, if we have a finite integral with a complex-valued leading order in the ϵ expansion, we must necessarily have a pole in ϵ in the total negative contribution which generates an imaginary ϵ^0 term when multiplied with the expansion of $(-1 - i\delta)^{-(\nu - LD/2)}$. In order for the full integral to be finite, the total positive contribution must have the exact same pole in ϵ such that the poles cancel in the full integral to leave a finite leading order.

4.2.2. Algorithm for univariate bisectable integrals

Here we describe an algorithmic procedure for resolving a class of Feynman integrals which we call *univariate bisectable (UB) in \mathbf{s}_R* . The goal of the algorithm is to map an integral in a given mixed-sign (Minkowski) regime, \mathbf{s}_R , defined by a set of inequalities depending on external kinematics, to a single integral in which $\mathcal{F}(\mathbf{x}; \mathbf{s})$ is non-negative and a single integral in which $\mathcal{F}(\mathbf{x}; \mathbf{s})$ is non-positive if there is a single variable for which the $\mathcal{F}(\mathbf{x}; \mathbf{s}) = 0$ hypersurface divides the integration domain in two. In Sec. 4.3.1 the algorithm is applied successfully to several non-trivial examples. In Sec. 4.3.2 we also solve cases for which this algorithm is not sufficient.

We begin by considering a generic Feynman integral of the form

$$J(\mathbf{s}) = \frac{(-1)^\nu \Gamma(\nu - LD/2)}{\prod_{i=1}^N \Gamma(\nu_i)} \lim_{\delta \rightarrow 0^+} I(\mathbf{s}; \delta), \quad (4.4)$$

$$\begin{aligned} I(\mathbf{s}; \delta) &= \int_{\mathbb{R}_{\geq 0}^N} d\mathbf{x} \mathcal{I}(\mathbf{x}; \mathbf{s}; \delta) \delta(1 - \alpha(\mathbf{x})) \\ &= \int_{\mathbb{R}_{\geq 0}^N} \prod_{i=1}^N \frac{dx_i}{x_i} x_i^{\nu_i} \frac{\mathcal{U}(\mathbf{x})^{\nu - (L+1)D/2}}{(\mathcal{F}(\mathbf{x}; \mathbf{s}) - i\delta)^{\nu - LD/2}} \delta(1 - \alpha(\mathbf{x})). \end{aligned} \quad (4.5)$$

with $d\mathbf{x} = \prod_{i=1}^N dx_i$. Our goal is to cast the integral into the form of Eq. (4.2) for a specific kinematic region.

We begin by defining some convenient notation. Let $\mathbf{x} = \{x_1, \dots, x_N\}$ be the complete set of Feynman parameters and $\mathbf{x}_{\neq i} = \mathbf{x} \setminus \{x_i\} = \{x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_N\}$ be the set excluding a single parameter x_i . Let $\mathbf{s}_R = \{\mathbf{s}_{\min} < \mathbf{s} < \mathbf{s}_{\max}\}$ be a kinematic region defined by a system of inequalities, this notation should be interpreted as placing minimum and maximum limits on each independent kinematic invariant/mass on which the integral depends (i.e. after applying momentum conservation to eliminate any dependent invariants). The success of the algorithm depends both on the Feynman integral and on the choice of kinematic region, a conservative choice of input region would be from one (pseudo-)threshold to the next (pseudo-)threshold in each variable without crossing any intermediate thresholds.

Algorithm 1 states the univariate bisection procedure. The input to the algorithm is the integrand of the Feynman integral to be resolved and the kinematic regime of interest. The algorithm iterates over each Feynman parameter and attempts to find a

Algorithm 1: Univariate Bisection (UB)

Input: $\mathcal{I}(\mathbf{x}; \mathbf{s}; \delta)$, \mathbf{s}_R
Output: $\mathcal{I}^+(\mathbf{x}; \mathbf{s})$, $\mathcal{I}^-(\mathbf{x}; \mathbf{s})$
foreach $x_i \in \mathbf{x}$ **do**
 Let $r = \text{Reduce}[\{\mathcal{F}(\mathbf{x}; \mathbf{s}) < 0\} \cup \{0 < \mathbf{x}\} \cup \mathbf{s}_R, x_i]$;
 if $r \sim \text{Eq. (4.6)}$ **then**
 Let $\mathcal{I}^-(\mathbf{x}; \mathbf{s}) = \mathcal{I}(\mathbf{x}_{\neq i}, y_i) \mathcal{I}(\mathbf{x}_{\neq i}, y_i; -\mathbf{s}; 0)$
 Let $\mathcal{I}^+(\mathbf{x}; \mathbf{s}) = \mathcal{I}'(\mathbf{x}_{\neq i}, y'_i) \mathcal{I}(\mathbf{x}_{\neq i}, y'_i; \mathbf{s}; 0)$
 return $\mathcal{I}^+(\mathbf{x}; \mathbf{s})$, $\mathcal{I}^-(\mathbf{x}; \mathbf{s})$
 else if $r \sim \text{Eq. (4.7)}$ **then**
 Let $\mathcal{I}^-(\mathbf{x}; \mathbf{s}) = \mathcal{I}'(\mathbf{x}_{\neq i}, y'_i) \mathcal{I}(\mathbf{x}_{\neq i}, y'_i; -\mathbf{s}; 0)$
 Let $\mathcal{I}^+(\mathbf{x}; \mathbf{s}) = \mathcal{I}(\mathbf{x}_{\neq i}, y_i) \mathcal{I}(\mathbf{x}_{\neq i}, y_i; \mathbf{s}; 0)$
 return $\mathcal{I}^+(\mathbf{x}; \mathbf{s})$, $\mathcal{I}^-(\mathbf{x}; \mathbf{s})$
end
return $\neg \text{UB}$ in \mathbf{s}_R

valid bisection. The **Reduce** procedure attempts to solve the system of inequalities in the variable x_i and is implemented, for example, in the **Mathematica** computer algebra system. The forms of the reduced system, r , for which a bisection is valid are given by either of

$$\{0 < x_i < f(\mathbf{x}_{\neq i})\} \quad \cup \quad \{0 < \mathbf{x}_{\neq i}\} \quad \cup \quad \mathbf{s}_R, \quad (4.6)$$

$$\{f(\mathbf{x}_{\neq i}) < x_i\} \quad \cup \quad \{0 < \mathbf{x}_{\neq i}\} \quad \cup \quad \mathbf{s}_R, \quad (4.7)$$

where $f(\mathbf{x}_{\neq i})$ is a rational (if $\mathcal{F}(\mathbf{x}; \mathbf{s})$ is linear in x_i) or, in general, an algebraic (if $\mathcal{F}(\mathbf{x}; \mathbf{s})$ is quadratic in x_i) function with unit degree of homogeneity. If a valid bisection can be found then we construct a transformation for the bisection parameter, x_i , that maps the variety to an integration boundary. If the reduced system r is of form Eq. (4.6) then we map $\mathcal{F} = 0$ to $x_i \rightarrow \infty$ while keeping the boundary at $x_i = 0$ fixed. If r is of the form Eq. (4.7) then we instead map the variety $\mathcal{F} = 0$ to $x_i = 0$ while keeping the boundary at $x_i \rightarrow \infty$ fixed. These mappings can be achieved by substituting x_i with y_i or y'_i , given by

$$y_i = \frac{x_i}{x_i + x_j} f(\mathbf{x}_{\neq i}), \quad (4.8)$$

$$y'_i = x_i + f(\mathbf{x}_{\neq i}). \quad (4.9)$$

These transformations are derived by solving $x'_i = \frac{x_i x_j}{f(\mathbf{x}_{\neq i}) - x_i}$ (for Eq. (4.8)) and $x'_i = x_i - f(\mathbf{x}_{\neq i})$ (for Eq. (4.9)) for x_i and then relabelling x'_i back to x_i post-transformation. In the mapping of Eq. (4.8) the variable $x_j \neq x_i$ appearing in the denominator is an arbitrary Feynman parameter. The function $\mathcal{J}(\mathbf{x}_{\neq i}, y_i)$ appearing in the algorithm is the Jacobian determinant resulting from the change of variables from x_i to y_i (similarly for $\mathcal{J}'(\mathbf{x}_{\neq i}, y'_i) = 1$). When defining $\mathcal{I}^-(\mathbf{x}; \mathbf{s})$ we factor a minus sign out of the \mathcal{F} -polynomial, we indicate this in our algorithm by calling \mathcal{I} with argument $-\mathbf{s}$. This is valid since only \mathcal{F} depends on the kinematics and it is linear in the squared invariants.

If the algorithm succeeds then it will return the non-negative integrands $\mathcal{I}^+(\mathbf{x}; \mathbf{s})$ and $\mathcal{I}^-(\mathbf{x}; \mathbf{s})$, the result for the original integral is then given by,

$$J^\pm(\mathbf{s}) = \frac{(-1)^\nu \Gamma(\nu - LD/2)}{\prod_{i=1}^N \Gamma(\nu_i)} \int_{\mathbb{R}_{\geq 0}^N} d\mathbf{x} \mathcal{I}^\pm(\mathbf{x}; \mathbf{s}) \delta(1 - \alpha^\pm(\mathbf{x})) \quad (4.10)$$

$$J(\mathbf{s}) = J^+(\mathbf{s}) + \lim_{\delta \rightarrow 0^+} (-1 - i\delta)^{-(\nu - LD/2)} J^-(\mathbf{s}). \quad (4.11)$$

If the algorithm fails then the integral is not UB in the kinematic region \mathbf{s}_R . A different (more restrictive) choice of region may be necessary, or the structure of the integral itself may prevent any univariate bisection from being obtained, we comment on this case further in Sec. 4.2.3.

Several aspects of the above algorithm are arbitrary and the outcome often depends on the exact ordering of Feynman parameters in \mathbf{x} . Firstly, there may be multiple possible bisection parameters for a given Feynman integral; in order to obtain the simplest possible integrands, it may be beneficial to examine multiple resolutions with different bisection parameters. It is sometimes even possible to select different bisection parameters for constructing $\mathcal{I}^-(\mathbf{x}; \mathbf{s})$ and $\mathcal{I}^+(\mathbf{x}; \mathbf{s})$. Secondly, the x_j appearing in Eq. (4.8) can be replaced by a more general function, e.g. a constant, however, we prefer the transformation to be homogeneous such that the resulting integrand retains its homogeneity. Any valid linear function of the Feynman parameters would also achieve this.

The univariate bisection algorithm is agnostic to the choice of hypersurface $\alpha(\mathbf{x})$ in the δ -function of Eq. (4.5), such that a choice of $\alpha^\pm(\mathbf{x})$ in Eq. (4.10) can be made post-resolution. However, we must ensure transformations are constructed such that some initial δ -functional that bounds the integral in the positive \mathbf{x} domain can be chosen. If the homogeneous transformation given in Eq. (4.8) is used, the integral remains projective and the existence of a valid initial δ -functional is automatically guaranteed (distinct choices of $\alpha^+(\mathbf{x})$ and $\alpha^-(\mathbf{x})$ may then be made, if so desired). This follows by viewing the original integral as defined over $\mathbb{RP}_{\geq 0}^{N-1}$ with the regions induced by bisection interpreted as projective subsets of $\mathbb{RP}_{\geq 0}^{N-1}$. This structure is preserved because the regions are bounded by coordinate hyperplanes and the projective variety defined by the zero locus of the homogeneous \mathcal{F} -polynomial. After being mapped back to $\mathbb{RP}_{\geq 0}^{N-1}$ via homogeneous transformations, each resulting region supports a well-defined projective integral which can be expressed as an integral over $\mathbb{R}_{\geq 0}^N$ with any standard choice of δ -functional constraint, such as the N -dimensional simplex or other admissible choices. In the numerical profiling of the examples of Sec. 4.3.1, we will choose $\delta(1 - x_N)$, with x_N the last Feynman parameter. In contrast to this, for the integrals which can not be bisected in a single variable, presented in Sec. 4.3.2, we will often make a specific choice of the δ -functional and integrate over one variable before considering the dissection of the integral.

4.2.3. Beyond univariate bisectable integrals

As is shown in Sec. 4.3.1, Algorithm 1 is sufficient to resolve various integrals at the 1- 2- and 3-loop level in kinematic regimes of physical interest. There is no particular obstruction of the algorithm related to the number of loops or legs present in an integral. With this said, there does exist simple examples for which the algorithm fails, one such example is provided by the 2-loop 4-point *planar* double-box integral in the regime

$\mathbf{s}_R = \{0 < s_{12} < \infty, -s_{12} < s_{23} < 0\}$, which is not univariate bisectable, unlike its non-planar cousin BNP7 (see Sec. 4.3.1) which is UB. One immediate generalisation of Algorithm 1, which has been verified to work on several examples, is to iterate the bisection using several variables until the integral is resolved. The primary complication of this procedure is ensuring that the entire original integration domain is covered without any double-counting. This complication motivates the general decomposition formula of Eq. (4.2) in which we the total number of positive and negative resolutions are not specified.

In Sec. 4.3.2, Feynman integrals involving internal massive propagators are examined. As mentioned previously, the $\mathcal{F}(\mathbf{x}; \mathbf{s})$ polynomial becomes quadratic in the Feynman parameters associated with the massive propagators, and it may not be possible to *bisect* the Feynman integral in such parameters (although there are massive examples which are UB with respect to a massive parameter). We show for individual integrals involving internal masses, including integrals known to be elliptic and hyperelliptic, that despite the failure of the UB algorithm in these cases, the integrals can nevertheless be resolved at the expense of introducing, in general, *algebraic* transformations related to the roots of a quadratic equation with coefficients generally depending on Feynman parameters and kinematic invariants. The goal of this chapter is to demonstrate that a resolution of the form in Eq. (4.2) is a desirable representation of a Feynman integral and that it can be achieved in a wide variety of non-trivial cases (massless/massive/non-planar and with various underlying integral geometries). A general solution to the problem of finding a resolution is given by a generic (or open) cylindrical algebraic decomposition [194], for which constructive algorithms exist.

4.3. Examples

4.3.1. Massless examples

In this section, we provide examples of massless integrals which are resolved by the algorithm presented in Sec. 4.2.2. We show that this procedure can be successfully applied to several integrals with up to 3 loops (including non-planar integrals) and up to 5 legs. The first example is the fully on-shell, 1-loop massless box. We apply the algorithm and carry out the steps in great detail to clarify the resolution procedure. The presentation of the remaining massless examples is more compact as they are resolved in the same way.

1-Loop Box with On-Shell Legs

The 1-loop massless box with all on-shell legs is shown in Fig. 4.2(a). The integral can be written in Feynman parameter space as,

$$J_{\text{box}}(\mathbf{s}) = \Gamma(2 + \epsilon) \lim_{\delta \rightarrow 0^+} I_{\text{box}}(\mathbf{s}; \delta), \quad (4.12)$$

$$I_{\text{box}}(\mathbf{s}; \delta) = \int_{\mathbb{R}_{\geq 0}^4} \prod_{i=1}^4 dx_i \frac{\mathcal{U}(\mathbf{x})^{2\epsilon}}{(\mathcal{F}(\mathbf{x}; \mathbf{s}) - i\delta)^{2+\epsilon}} \delta(1 - \alpha(\mathbf{x})), \quad (4.13)$$

where the $\mathcal{U}(\mathbf{x})$ and $\mathcal{F}(\mathbf{x}; \mathbf{s})$ polynomials are given by

$$\mathcal{U}(\mathbf{x}) = x_1 + x_2 + x_3 + x_4, \quad (4.14)$$

$$\mathcal{F}(\mathbf{x}; \mathbf{s}) = -s_{12}x_1x_2 - s_{13}x_3x_4, \quad (4.15)$$

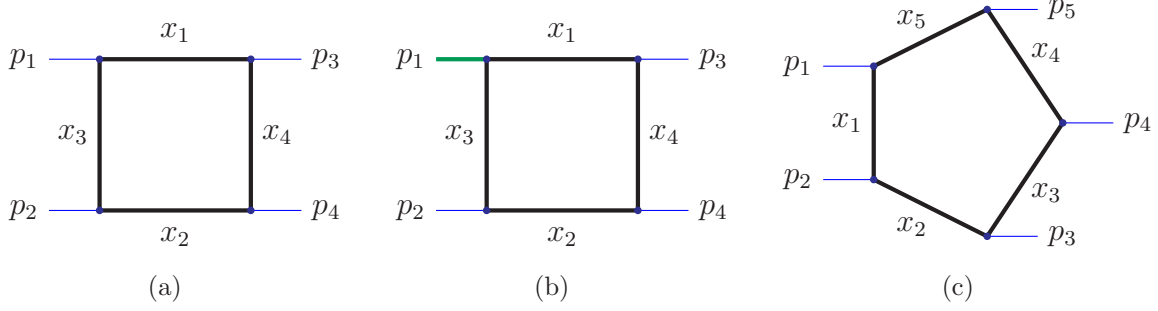


Figure 4.2.: The massless box with all on-shell legs (4.2(a)), an off-shell leg (p_1) (4.2(b)) and the massless pentagon (4.2(c)).

with $s_{ij} = (p_i + p_j)^2$. Suppose that we are interested in evaluating this integral for $2 \rightarrow 2$ physical scattering kinematics $\mathbf{s}_{\text{phys}} = \{0 < s_{12} < \infty, -s_{12} < s_{13} < 0\}$. Usually we would need a contour deformation as this regime is above the threshold at $s_{12} = 0$ and the \mathcal{F} polynomial contains monomials of different sign and can itself be both positive or negative inside the domain of integration. Recall that in this kinematic regime $s_{12} > 0$ and $(-s_{13}) > 0$.

Applying Algorithm 1, we begin with the choice $x_i = x_1$. Attempting to reduce with respect to x_1 the set of inequalities,

$$\{-s_{12}x_1x_2 - s_{13}x_3x_4 < 0\} \cup \{0 < x_1, 0 < x_2, 0 < x_3, 0 < x_4\} \cup \mathbf{s}_{\text{phys}}, \quad (4.16)$$

we obtain the solution,

$$r = \left\{ \frac{-s_{13}x_3x_4}{s_{12}x_2} < x_1 \right\} \cup \{0 < x_2, 0 < x_3, 0 < x_4\} \cup \mathbf{s}_{\text{phys}}. \quad (4.17)$$

We observe that r is of the form given in Eq. (4.7) with

$$f(\mathbf{x}_{\neq 1}) = \frac{(-s_{13})x_3x_4}{s_{12}x_2}, \quad (4.18)$$

therefore, x_1 is a valid bisection parameter. As dictated by the algorithm, we can now construct the positive and negative contributions by transforming the variable x_1 . Applying the transformation given in Eq. (4.7),

$$x_1 \rightarrow y'_1 = x_1 + \frac{(-s_{13})x_3x_4}{s_{12}x_2}, \quad (4.19)$$

we map the variety $\mathcal{F}(\mathbf{x}; \mathbf{s}) = 0$ to $x_1 = 0$ while keeping the boundary at $x_1 \rightarrow \infty$ fixed. The resulting integrand is given by

$$\mathcal{J}^-(\mathbf{x}) = 1, \quad \mathcal{U}^-(\mathbf{x}) = x_1 + x_2 + x_3 + x_4 + \frac{-s_{13}x_3x_4}{s_{12}x_2}, \quad \mathcal{F}^-(\mathbf{x}; \mathbf{s}) = s_{12}x_1x_2, \quad (4.20)$$

$$\begin{aligned} \mathcal{I}_{\text{box}}^-(\mathbf{x}; \mathbf{s}) &= \mathcal{J}^-(\mathbf{x}) \frac{\mathcal{U}^-(\mathbf{x})^{2\epsilon}}{\mathcal{F}^-(\mathbf{x}; \mathbf{s})^{2+\epsilon}} \\ &= x_1^{-2-\epsilon} (s_{12}x_2)^{-2-3\epsilon} (s_{12}x_2(x_1 + x_2 + x_3 + x_4) - s_{13}x_3x_4)^{2\epsilon}. \end{aligned} \quad (4.21)$$

In this example we are not considering an integral with dots (propagators raised to a higher power) or with a numerator. In general one would apply the transformation also to any Feynman parameters appearing in the numerator.

The positive contribution is given by transforming the variable x_1 according to Eq. (4.8) with the arbitrary choice $x_j = x_N = x_4$,

$$x_1 \rightarrow y_1 = \frac{x_1}{x_1 + x_4} \frac{(-s_{13})x_3x_4}{s_{12}x_2} \quad (4.22)$$

which maps the variety $\mathcal{F}(\mathbf{x}; \mathbf{s}) = 0$ to $x_1 \rightarrow \infty$ keeping $x_1 = 0$ fixed. The resulting integrand is given by

$$\mathcal{J}^+(\mathbf{x}) = \frac{(-s_{13})x_3x_4^2}{s_{12}x_2(x_1 + x_4)^2}, \quad \mathcal{U}^+(\mathbf{x}) = \frac{x_1}{x_1 + x_4} \frac{(-s_{13})x_3x_4}{s_{12}x_2} + x_2 + x_3 + x_4, \quad (4.23)$$

$$\mathcal{F}^+(\mathbf{x}; \mathbf{s}) = \frac{(-s_{13})x_3x_4^2}{x_1 + x_4}, \quad (4.24)$$

$$\begin{aligned} \mathcal{I}_{\text{box}}^+(\mathbf{x}; \mathbf{s}) &= \mathcal{J}^+(\mathbf{x}) \frac{\mathcal{U}^+(\mathbf{x})^{2\epsilon}}{\mathcal{F}^+(\mathbf{x}; \mathbf{s})^{2+\epsilon}} \\ &= (x_1 + x_4)^{-\epsilon} (s_{12}x_2)^{-1-2\epsilon} (-s_{13}x_3x_4^2)^{-1-\epsilon} \\ &\quad (s_{12}x_2 (x_1 + x_4) (x_2 + x_3 + x_4) - s_{13}x_1x_3x_4)^{2\epsilon}. \end{aligned} \quad (4.25)$$

Sewing the positive and negative contributions together, the final result for the on-shell box integral is given by,

$$J_{\text{box}}(\mathbf{s}) = \Gamma(2 + \epsilon) \lim_{\delta \rightarrow 0^+} I_{\text{box}}(\mathbf{s}; \delta) \quad (4.26)$$

$$I_{\text{box}}(\mathbf{s}; \delta) = I_{\text{box}}^+(\mathbf{s}) + (-1 - i\delta)^{-2-\epsilon} I_{\text{box}}^-(\mathbf{s}). \quad (4.27)$$

Neglecting the δ -functions, the integrands of both I_{box}^+ and I_{box}^- pick up a factor of $\lambda^{-N} = \lambda^{-4}$ under the scaling transformation $\{x_1, \dots, x_4\} \rightarrow \{\lambda x_1, \dots, \lambda x_4\}$. This stems from the homogeneity of the original \mathcal{U} and \mathcal{F} polynomials and the fact that the algorithm preserves the homogeneity.

1-Loop Box with an Off-Shell Leg

The procedure above is straightforwardly extended to the massless box with an off-shell leg, $p_1^2 > 0$, shown in Fig. 4.2(b). The integral is

$$J_{\text{box}, p_1^2 > 0}(\mathbf{s}) = \Gamma(2 + \epsilon) \lim_{\delta \rightarrow 0^+} I_{\text{box}, p_1^2 > 0}(\mathbf{s}; \delta) \quad (4.28)$$

$$I_{\text{box}, p_1^2 > 0}(\mathbf{s}; \delta) = \int_{\mathbb{R}_{\geq 0}^4} \prod_{i=1}^4 dx_i \frac{\mathcal{U}(\mathbf{x})^{2\epsilon}}{(\mathcal{F}(\mathbf{x}; \mathbf{s}) - i\delta)^{2+\epsilon}} \delta(1 - \alpha(\mathbf{x})). \quad (4.29)$$

The \mathcal{F} polynomial is now

$$\mathcal{F}(\mathbf{x}; \mathbf{s}) = -s_{12}x_1x_2 - s_{13}x_3x_4 - p_1^2x_1x_3. \quad (4.30)$$

and $\mathcal{U}(\mathbf{x})$ is unchanged. The integral is again UB with respect to x_1 in $\mathbf{s}_{p_1^2 > 0} = \{0 < p_1^2 < \infty, 0 < s_{12} < \infty, -s_{12} < s_{13} < 0\}$. Choosing again $x_j = x_4$ the transformations are

$$x_1 \rightarrow y'_1 = x_1 + \left(\frac{-s_{13}x_3x_4}{s_{12}x_2 + p_1^2x_3} \right), \quad \mathcal{F}^-(\mathbf{x}; \mathbf{s}) = x_1(s_{12}x_2 + p_1^2x_3), \quad (4.31)$$

$$x_1 \rightarrow y_1 = \frac{x_1}{x_1 + x_4} \left(\frac{-s_{13}x_3x_4}{s_{12}x_2 + p_1^2x_3} \right), \quad \mathcal{F}^+(\mathbf{x}; \mathbf{s}) = \frac{-s_{13}x_3x_4^2}{x_1 + x_4}. \quad (4.32)$$

Applying the transformations to $\mathcal{U}(\mathbf{x})$ as well as including the Jacobians we obtain the resolved integrands

$$\mathcal{I}_{\text{box}, p_1^2 > 0}^-(\mathbf{x}; \mathbf{s}) = \frac{[(s_{12}x_2 + p_1^2x_3)(x_1 + x_2 + x_3 + x_4) - s_{13}x_3x_4]^{2\epsilon}}{x_1^{2+\epsilon}(s_{12}x_2 + p_1^2x_3)^{2+3\epsilon}}, \quad (4.33)$$

$$\mathcal{I}_{\text{box}, p_1^2 > 0}^+(\mathbf{x}; \mathbf{s}) = \frac{[(s_{12}x_2 + p_1^2x_3)(x_1 + x_4)(x_2 + x_3 + x_4) - s_{13}x_1x_3x_4]^{2\epsilon}}{x_4^{2+2\epsilon}(x_1 + x_4)^\epsilon(-s_{13}x_3)^{1+\epsilon}(s_{12}x_2 + p_1^2x_3)^{1+2\epsilon}}. \quad (4.34)$$

The final result for the off-shell box integral is given by,

$$J_{\text{box}, p_1^2 > 0}(\mathbf{s}) = \Gamma(2 + \epsilon) \lim_{\delta \rightarrow 0^+} I_{\text{box}, p_1^2 > 0}(\mathbf{s}; \delta), \quad (4.35)$$

$$I_{\text{box}, p_1^2 > 0}(\mathbf{s}; \delta) = I_{\text{box}, p_1^2 > 0}^+(\mathbf{s}) + (-1 - i\delta)^{-2-\epsilon} I_{\text{box}, p_1^2 > 0}^-(\mathbf{s}). \quad (4.36)$$

1-Loop On-Shell Pentagon

Now we come to the example from the introduction, the massless one-loop pentagon, seen in Fig. 4.2(c). The pentagon is minimally parameterised with five kinematic invariants, which for example can be the set of cyclic scalar products $(s_{12}, s_{23}, s_{34}, s_{45}, s_{51})$, where $s_{ij} = (p_i + p_j)^2$. The integral is

$$J_{\text{pen}}(\mathbf{s}) = -\Gamma(3 + \epsilon) \lim_{\delta \rightarrow 0^+} I_{\text{pen}}(\mathbf{s}; \delta) \quad (4.37)$$

$$I_{\text{pen}}(\mathbf{s}; \delta) = \int_{\mathbb{R}_{\geq 0}^5} \prod_{i=1}^5 dx_i \frac{\mathcal{U}(\mathbf{x})^{1+2\epsilon}}{(\mathcal{F}(\mathbf{x}; \mathbf{s}) - i\delta)^{3+\epsilon}} \delta \left(1 - \sum_{i=1}^5 \alpha_i x_i \right) \quad (4.38)$$

and the graph polynomials are

$$\mathcal{U}(\mathbf{x}) = x_1 + x_2 + x_3 + x_4 + x_5 \quad (4.39)$$

$$\mathcal{F}(\mathbf{x}; \mathbf{s}) = -s_{12}x_2x_5 - s_{23}x_1x_3 - s_{34}x_2x_4 - s_{45}x_3x_5 - s_{51}x_1x_4. \quad (4.40)$$

We arbitrarily select a mixed-sign kinematic regime given by $\mathbf{s}_R = \{0 < s_{12}, s_{34}, s_{51} < \infty, -\infty < s_{23}, s_{45} < 0\}$. The integral turns out to be UB with respect to x_3 . Defining

$$f(\mathbf{x}_{\neq 3}) = \frac{s_{51}x_1x_4 + s_{34}x_2x_4 + s_{12}x_2x_5}{-s_{23}x_1 - s_{45}x_5} \quad (4.41)$$

the transformations (with $x_j = x_5$) are given by

$$x_3 \rightarrow y_3 = \frac{x_3}{x_3 + x_5} f(\mathbf{x}_{\neq 3}), \quad \mathcal{F}^-(\mathbf{x}; \mathbf{s}) = \frac{x_5(s_{51}x_1x_4 + s_{34}x_2x_4 + s_{12}x_2x_5)}{x_3 + x_5}, \quad (4.42)$$

$$x_3 \rightarrow y'_3 = x_3 + f(\mathbf{x}_{\neq 3}), \quad \mathcal{F}^+(\mathbf{x}; \mathbf{s}) = -x_3(s_{23}x_1 + s_{45}x_5). \quad (4.43)$$

Applying them also to $\mathcal{U}(\mathbf{x})$ and including the Jacobians, the resulting integrands are

$$\begin{aligned} \mathcal{I}_{\text{pen}}^- &= x_5^{-2-\epsilon} (x_3 + x_5)^{-\epsilon} (-s_{23}x_1 - s_{45}x_5)^{-2-2\epsilon} (s_{51}x_1x_4 + s_{34}x_2x_4 + s_{12}x_2x_5)^{-2-\epsilon} \times \\ &\quad [(x_3 + x_5)(-s_{23}x_1 - s_{45}x_5)(x_1 + x_2 + x_3 + x_4 + x_5) + \\ &\quad x_3(s_{51}x_1x_4 + s_{34}x_2x_4 + s_{12}x_2x_5)]^{1+2\epsilon}, \end{aligned} \quad (4.44)$$

$$\begin{aligned} \mathcal{I}_{\text{pen}}^+ &= x_3^{-3-\epsilon} (-s_{23}x_1 - s_{45}x_5)^{-4-3\epsilon} \times \\ &\quad [(-s_{23}x_1 - s_{45}x_5)(x_1 + x_2 + x_3 + x_4 + x_5) + s_{51}x_1x_4 + s_{34}x_2x_4 + s_{12}x_2x_5]^{1+2\epsilon}. \end{aligned} \quad (4.45)$$

Summing the positive and negative contributions yields the resolved pentagon integral,

$$J_{\text{pen}}(\mathbf{s}) = -\Gamma(3 + \epsilon) \lim_{\delta \rightarrow 0^+} I_{\text{pen}}(\mathbf{s}; \delta), \quad (4.46)$$

$$I_{\text{pen}}(\mathbf{s}; \delta) = I_{\text{pen}}^+(\mathbf{s}) + (-1 - i\delta)^{-3-\epsilon} I_{\text{pen}}^-(\mathbf{s}). \quad (4.47)$$

The resolution described above is valid for one particular sign combination of the kinematics, \mathbf{s}_R . To evaluate the pentagon for any kinematic point, one would naively have to repeat the procedure above $2^5 = 32$ times. Due to reparametrisation invariance, however, only 4 resolution procedures are required, corresponding to the presence of 0, 1 or 2 negative invariants, where the case with 2 negative invariants requires different resolutions depending on whether the legs with negative invariants are adjacent or not. For example, the regime where $-\infty < s_{12}, s_{34}, s_{51} < 0 < s_{23}, s_{45} < \infty$ can be directly related to the case described above by factoring out a minus sign from \mathcal{F} . For completeness, we highlight that the UB algorithm is sufficient to resolve kinematic regions where either 2 or 3 invariants are negative, and where the propagators that connect the legs corresponding to each of the same sign invariants are not adjacent to each other. In other configurations, the space is only dissected into positive and negative parts with constraints on more than one Feynman parameter.

If we wish to compute 1-loop integrals with more than 5 legs, we can continue to derive resolutions using the techniques described above. However, all dimensionally regulated 1-loop N -point integrals can be reduced at all orders in ϵ to integrals with at most 5 legs [85, 195–198], therefore, it is sufficient to resolve all 1-loop integrals up to and including pentagons. Furthermore, if we wish to evaluate only up to and including the finite order in ϵ , we can reduce all 1-loop integrals with 5 or more legs to box integrals with off-shell legs and resolve these instead. For example, the 1-loop pentagon integral can be written in terms of boxes as [195, 196],

$$I_{\text{pen}} = -\frac{1}{2} \sum_{l,k=1}^5 S_{lk}^{-1} I_{\text{box},l} + \mathcal{O}(\epsilon), \quad (4.48)$$

where S_{lk} are elements of the S-matrix

$$S = -\frac{1}{2} \begin{pmatrix} 0 & p_2^2 & s_{23} & s_{51} & p_1^2 \\ p_2^2 & 0 & p_3^2 & s_{34} & s_{12} \\ s_{23} & p_3^2 & 0 & p_4^2 & s_{45} \\ s_{51} & s_{34} & p_4^2 & 0 & p_5^2 \\ p_1^2 & s_{12} & s_{45} & p_5^2 & 0 \end{pmatrix}. \quad (4.49)$$

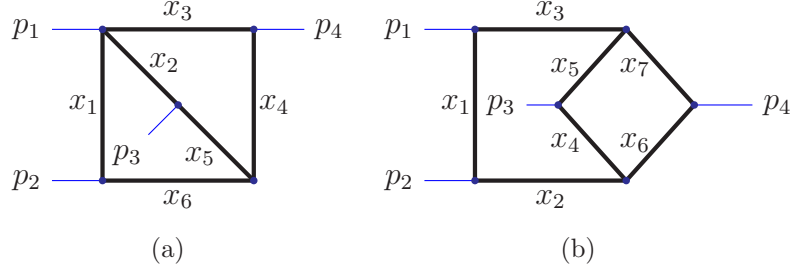


Figure 4.3.: Massless non-planar 2-loop boxes with 6 (Fig. 4.3(a): BNP6) and 7 (Fig. 4.3(b): BNP7) propagators respectively.

In Eq. (4.48), each box integral has one off-shell leg corresponding to the pinching of a propagator of the pentagon.

2-Loop Non-Planar 6-Propagator Box

The non-planar box with six propagators (BNP6, see Fig. 4.3(a)) can be parameterised by the Mandelstam invariants $s_{12} = (p_1 + p_2)^2$ and $s_{23} = (p_2 + p_3)^2$ after applying the momentum conservation rule $s_{12} + s_{23} + s_{13} = 0$ to eliminate s_{13} . The integral may be written as,

$$J_{\text{BNP6}}(\mathbf{s}) = \Gamma(2 + 2\epsilon) \lim_{\delta \rightarrow 0^+} I_{\text{BNP6}}(\mathbf{s}; \delta) \quad (4.50)$$

$$I_{\text{BNP6}}(\mathbf{s}; \delta) = \int_{\mathbb{R}_{\geq 0}^6} \prod_{i=1}^6 dx_i \frac{\mathcal{U}(\mathbf{x})^{3\epsilon}}{(\mathcal{F}(\mathbf{x}; \mathbf{s}) - i\delta)^{2+2\epsilon}} \delta(1 - \alpha(\mathbf{x})) \quad (4.51)$$

with the \mathcal{U} and \mathcal{F} polynomials,

$$\mathcal{U}(\mathbf{x}) = x_1x_2 + x_1x_3 + x_1x_4 + x_1x_5 + x_2x_3 + x_2x_4 + x_2x_6 + x_3x_5 + x_3x_6 + x_4x_5 + x_4x_6 + x_5x_6, \quad (4.52)$$

$$\mathcal{F}(\mathbf{x}; \mathbf{s}) = -s_{12}x_2x_3x_6 - s_{23}x_1x_2x_4 + (s_{12} + s_{23})x_1x_3x_5. \quad (4.53)$$

In the physical kinematic regime for massless $2 \rightarrow 2$ scattering, $\mathbf{s}_{\text{phys}} = \{0 < s_{12} < \infty, -s_{12} < s_{23} < 0\}$, the integral is UB with respect with x_1 . Choosing $x_j = x_6$ the transformations are

$$x_1 \rightarrow y_1 = \frac{x_1}{x_1 + x_6} f(\mathbf{x}_{\neq 1}), \quad \mathcal{F}^-(\mathbf{x}; \mathbf{s}) = \frac{s_{12}x_2x_3x_6^2}{x_1 + x_6}, \quad (4.54)$$

$$x_1 \rightarrow y'_1 = x_1 + f(\mathbf{x}_{\neq 1}), \quad \mathcal{F}^+(\mathbf{x}; \mathbf{s}) = x_1 [(s_{12} + s_{23})x_3x_5 - s_{23}x_2x_4], \quad (4.55)$$

with

$$f(\mathbf{x}_{\neq 1}) = \frac{s_{12}x_2x_3x_6}{(s_{12} + s_{23})x_3x_5 - s_{23}x_2x_4}, \quad (4.56)$$

These transformations result in the resolved integrands,

$$\begin{aligned} \mathcal{I}_{\text{BNP6}}^- &= (s_{12}x_2x_3x_6^2)^{-1-2\epsilon} (x_1+x_6)^{-\epsilon} [(s_{12}+s_{23})x_3x_5-s_{23}x_2x_4]^{-1-3\epsilon} \times \\ &\quad \left[[(s_{12}+s_{23})x_3x_5-s_{23}x_2x_4] (x_1+x_6) [(x_3+x_4)(x_2+x_5)+ \right. \\ &\quad \left. (x_2+x_3+x_4+x_5)x_6] + s_{12}x_1x_2x_3x_6 (x_2+x_3+x_4+x_5) \right]^{3\epsilon}, \end{aligned} \quad (4.57)$$

$$\begin{aligned} \mathcal{I}_{\text{BNP6}}^+ &= x_1^{-2-2\epsilon} [(s_{12}+s_{23})x_3x_5-s_{23}x_2x_4]^{-2-5\epsilon} \times \\ &\quad \left[[(s_{12}+s_{23})x_3x_5-s_{23}x_2x_4] [(x_3+x_4)(x_2+x_5)+ \right. \\ &\quad \left. (x_1+x_6)(x_2+x_3+x_4+x_5)] + s_{12}x_2x_3x_6 (x_2+x_3+x_4+x_5) \right]^{3\epsilon}. \end{aligned} \quad (4.58)$$

This is an example of a resolved integrand which is not *manifestly* positive. Instead, it is the restrictions on the kinematics that ensure the positivity. For example, $s_{12}+s_{23}$ is positive since we assume $s_{12} > -s_{23}$. Combining everything, the resolved BNP6 integral is

$$J_{\text{BNP6}}(\mathbf{s}) = \Gamma(2+2\epsilon) \lim_{\delta \rightarrow 0^+} I_{\text{BNP6}}(\mathbf{s}; \delta) \quad (4.59)$$

$$I_{\text{BNP6}}(\mathbf{s}; \delta) = I_{\text{BNP6}}^+(\mathbf{s}) + (-1-i\delta)^{-2-2\epsilon} I_{\text{BNP6}}^-(\mathbf{s}). \quad (4.60)$$

2-Loop Non-Planar 7-Propagator Box

The non planar box with seven propagators (BNP7, see Fig. 4.3(b)) is parameterised in the same way as BNP6. The integral is given by

$$J_{\text{BNP7}} = -\Gamma(3+2\epsilon) \lim_{\delta \rightarrow 0^+} I_{\text{BNP7}} \quad (4.61)$$

$$I_{\text{BNP7}} = \int_{\mathbb{R}_{\geq 0}^7} \prod_{i=1}^7 dx_i \frac{\mathcal{U}(\mathbf{x})^{1+3\epsilon}}{(\mathcal{F}(\mathbf{x}; \mathbf{s}) - i\delta)^{3+2\epsilon}} \delta(1-\alpha(\mathbf{x})) \quad (4.62)$$

with the \mathcal{U} and \mathcal{F} polynomials,

$$\begin{aligned} \mathcal{U}(\mathbf{x}) &= x_1x_4 + x_1x_5 + x_1x_6 + x_1x_7 + x_2x_4 + x_2x_5 + x_2x_6 + x_2x_7 + \\ &\quad x_3x_4 + x_3x_5 + x_3x_6 + x_3x_7 + x_4x_6 + x_4x_7 + x_5x_6 + x_5x_7 \end{aligned} \quad (4.63)$$

$$\begin{aligned} \mathcal{F}(\mathbf{x}, \mathbf{s}) &= -s_{12}(x_3x_4x_6 + x_2x_5x_7 + x_2x_3x_7 + x_2x_3x_6 + x_2x_3x_5 + x_2x_3x_4) \\ &\quad - s_{23}x_1x_5x_6 + (s_{12}+s_{23})x_1x_4x_7. \end{aligned} \quad (4.64)$$

Similar to BNP6, this integral is UB with respect to x_1 in \mathbf{s}_{phys} . Choosing $x_j = x_7$:

$$x_1 \rightarrow y_1 = \frac{x_1}{x_1+x_7} f(\mathbf{x}_{\neq 1}), \quad \mathcal{F}^-(\mathbf{x}; \mathbf{s}) = \frac{s_{12}x_7 [x_3x_4x_6 + x_2x_5x_7 + x_2x_3(x_4+x_5+x_6+x_7)]}{x_1+x_7}, \quad (4.65)$$

$$x_1 \rightarrow y'_1 = x_1 + f(\mathbf{x}_{\neq 1}), \quad \mathcal{F}^+(\mathbf{x}; \mathbf{s}) = x_1 [(s_{12}+s_{23})x_4x_7 - s_{23}x_5x_6] \quad (4.66)$$

where

$$f(\mathbf{x}_{\neq 1}) = \frac{s_{12} [x_3x_4x_6 + x_2x_5x_7 + x_2x_3(x_4+x_5+x_6+x_7)]}{(s_{12}+s_{23})x_4x_7 - s_{23}x_5x_6} \quad (4.67)$$

The resulting integrands are

$$\begin{aligned} \mathcal{I}_{\text{BNP7}}^- &= (s_{12}x_7)^{-2-2\epsilon} (x_1 + x_7)^{-\epsilon} [(s_{12} + s_{23})x_4x_7 - s_{23}x_5x_6]^{-2-3\epsilon} \times \\ &\quad [x_3x_4x_6 + x_2x_5x_7 + x_2x_3(x_4 + x_5 + x_6 + x_7)]^{-2-2\epsilon} \times \\ &\quad \left[s_{12}x_1(x_4 + x_5 + x_6 + x_7) [x_3x_4x_6 + x_2x_5x_7 + x_2x_3(x_4 + x_5 + x_6 + x_7)] + \right. \\ &\quad \left. [(s_{12} + s_{23})x_4x_7 - s_{23}x_5x_6] (x_1 + x_7) [(x_4 + x_5)(x_6 + x_7) + \right. \\ &\quad \left. (x_2 + x_3)(x_4 + x_5 + x_6 + x_7)] \right]^{1+3\epsilon}, \end{aligned} \quad (4.68)$$

$$\begin{aligned} \mathcal{I}_{\text{BNP7}}^+ &= x_1^{-3-2\epsilon} [(s_{12} + s_{23})x_4x_7 - s_{23}x_5x_6]^{-4-5\epsilon} \times \\ &\quad \left[s_{12}(x_4 + x_5 + x_6 + x_7) [x_3x_4x_6 + x_2x_5x_7 + x_2x_3(x_4 + x_5 + x_6 + x_7)] + \right. \\ &\quad \left. [(s_{12} + s_{23})x_4x_7 - s_{23}x_5x_6] [(x_4 + x_5)x_6 + x_3(x_4 + x_5 + x_6) + \right. \\ &\quad \left. (x_3 + x_4 + x_5)x_7 + (x_1 + x_2)(x_4 + x_5 + x_6 + x_7)] \right]^{1+3\epsilon}. \end{aligned} \quad (4.69)$$

This is again not manifestly positive, but the assumptions on the kinematics ensures that the resolution is valid. Combining the positive and negative contributions gives the resolution of BNP7,

$$J_{\text{BNP7}}(\mathbf{s}) = -\Gamma(3 + 2\epsilon) \lim_{\delta \rightarrow 0^+} I_{\text{BNP7}}(\mathbf{s}; \delta) \quad (4.70)$$

$$I_{\text{BNP7}}(\mathbf{s}; \delta) = I_{\text{BNP7}}^+(\mathbf{s}) + (-1 - i\delta)^{-3-2\epsilon} I_{\text{BNP7}}^-(\mathbf{s}). \quad (4.71)$$

3-Loop Non-Planar Box

The final massless example is the massless 3-loop 4-point non-planar box ($G_{\bullet\bullet}$ in the notation of Ref. [81]) where a naive contour deformation in Feynman parameter space completely fails due to the presence of a leading Landau singularity within the domain of integration. We show that, after first dissecting the integral on the parameter space hypersurface associated with the Landau singularity, we can apply a combination of shifts, rescalings, and rational transformations of the Feynman parameters to resolve the mixed-sign integrals.

We refer to this example as the crown integral, see Fig. 4.4. Using the same parametrisation as in BNP6 and BNP7 the crown integral is

$$J_{G_{\bullet\bullet}}(\mathbf{s}) = \Gamma(2 + 3\epsilon) \lim_{\delta \rightarrow 0^+} I_{G_{\bullet\bullet}}(\mathbf{s}; \delta) \quad (4.72)$$

$$I_{G_{\bullet\bullet}}(\mathbf{s}; \delta) = \int_{\mathbb{R}_{\geq 0}^8} \prod_{i=1}^8 dx_i \frac{\mathcal{U}(\mathbf{x})^{4\epsilon}}{(\mathcal{F}(\mathbf{x}, \mathbf{s}) - i\delta)^{2+3\epsilon}} \delta(1 - \alpha(\mathbf{x})) \quad (4.73)$$

where the \mathcal{U} and \mathcal{F} polynomials are

$$\begin{aligned} \mathcal{U}(\mathbf{x}) &= (x_1 + x_2)(x_3 + x_4)(x_5 + x_6) + (x_1 + x_2)(x_3 + x_4)(x_7 + x_8) + \\ &\quad (x_1 + x_2)(x_5 + x_6)(x_7 + x_8) + (x_3 + x_4)(x_5 + x_6)(x_7 + x_8), \end{aligned} \quad (4.74)$$

$$\mathcal{F}(\mathbf{x}, \mathbf{s}) = -s_{12}(x_2x_5 - x_1x_6)(x_4x_7 - x_3x_8) - s_{13}(x_2x_3 - x_1x_4)(x_6x_7 - x_5x_8). \quad (4.75)$$

We restrict to the massless $2 \rightarrow 2$ physical scattering regime,

$$\mathbf{s}_{\text{phys}} = \{0 < s_{12} < \infty, -s_{12} < s_{13} < 0\}. \quad (4.76)$$

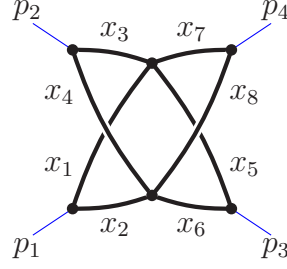


Figure 4.4.: Massless non-planar 3-loop box (the crown graph, $G_{\bullet\bullet}$).

In Ref. [81] it was shown that this integral has a solution of the Landau equations within the integration domain for generic physical kinematics, preventing evaluation with the usual contour deformation procedure. The solution proposed by the authors is to transform the Feynman parameters such that each factor of $\mathcal{F}(\mathbf{x}; \mathbf{s})$ is linear. This can be achieved by the transformations

$$x_1 \rightarrow x_1 \frac{x_2}{x_8}, \quad x_3 \rightarrow x_3 \frac{x_4}{x_8}, \quad x_5 \rightarrow x_5 \frac{x_6}{x_8}, \quad x_7 \rightarrow x_7 \frac{x_8}{x_8}, \quad (4.77)$$

with Jacobian,

$$\mathcal{J} = \frac{x_2 x_4 x_6}{x_8^3}. \quad (4.78)$$

The resulting \mathcal{F} -polynomial is given by,

$$\mathcal{F}(\mathbf{x}; \mathbf{s}) = \frac{x_2 x_4 x_6}{x_8} (-s_{12}(x_1 - x_5)(x_3 - x_7) - s_{13}(x_1 - x_3)(x_5 - x_7)). \quad (4.79)$$

The integral can then be dissected into $4! = 24$ regions each defined by a strict ordering of $x_i \geq x_j \geq x_k \geq x_l$ with $\{i, j, k, l\}$ all permutations of $\{1, 3, 5, 7\}$. Each of the resulting integrals will have polynomials of definite sign multiplying each invariant s_{12} or s_{13} and can in principle be evaluated using contour deformation. Taking into account the symmetry of the integral, $I_{G_{\bullet\bullet}}$ can now be expressed as a sum over six integrals,

$$I_{G_{\bullet\bullet}}(\mathbf{s}; \delta) = 4 \sum_K I_K(\mathbf{s}; \delta), \quad (4.80)$$

$$I_K(\mathbf{s}; \delta) = \int_{\mathbb{R}_{\geq 0}^8} \prod_{i=1}^8 dx_i \frac{\mathcal{U}_K(\mathbf{x})^{4\epsilon}}{(x_2 x_4 x_6)^{1+3\epsilon} x_8^{1+9\epsilon} (\mathcal{F}_K(\mathbf{x}, \mathbf{s}) - i\delta)^{2+3\epsilon}} \delta(1 - \alpha(\mathbf{x})), \quad (4.81)$$

where $K \in \{A, B, C, D, E, F\}$, denoting each of the six integrals resulting from the resolution of the Landau singularity. The \mathcal{F}_K polynomials are

$$\mathcal{F}_A(\mathbf{x}, \mathbf{s}) = -[s_{12}x_3(x_1 + x_3 + x_5) + (s_{12} + s_{13})x_1x_5] \quad (4.82)$$

$$\mathcal{F}_B(\mathbf{x}, \mathbf{s}) = -[(s_{12} + s_{13})x_1x_3 + s_{13}x_5(x_1 + x_3 + x_5)] \quad (4.83)$$

$$\mathcal{F}_C(\mathbf{x}, \mathbf{s}) = -[s_{12}x_1(x_1 + x_3 + x_5) - s_{13}x_3x_5] \quad (4.84)$$

$$\mathcal{F}_D(\mathbf{x}, \mathbf{s}) = +[(s_{12} + s_{13})x_5(x_1 + x_3 + x_5) + s_{13}x_1x_3] \quad (4.85)$$

$$\mathcal{F}_E(\mathbf{x}, \mathbf{s}) = +[s_{12}x_3x_5 - s_{13}x_1(x_1 + x_3 + x_5)] \quad (4.86)$$

$$\mathcal{F}_F(\mathbf{x}, \mathbf{s}) = +[s_{12}x_1x_5 + (s_{12} + s_{13})x_3(x_1 + x_3 + x_5)]. \quad (4.87)$$

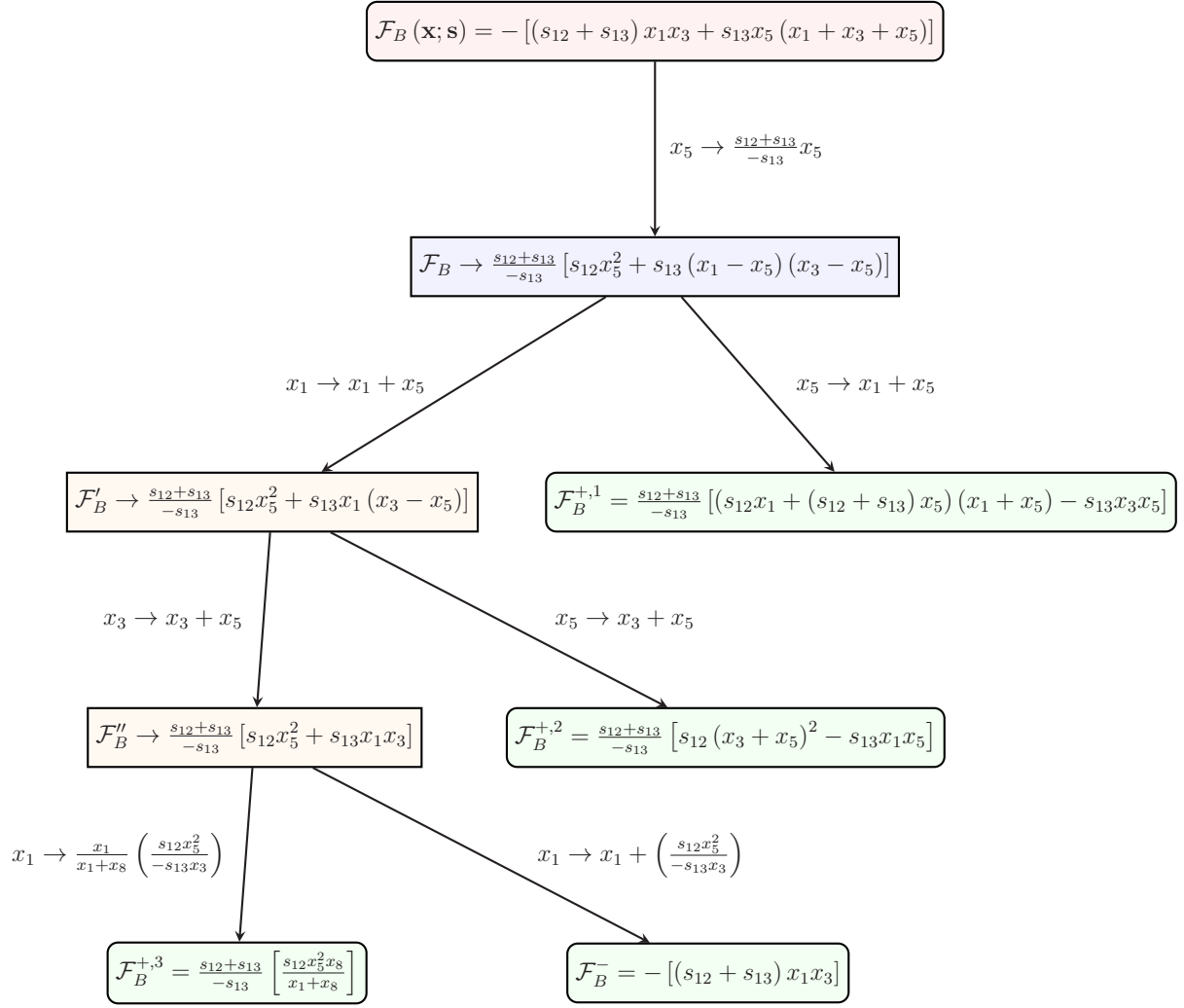


Figure 4.5.: Example transformations for resolving integral B . The initial integral (red) is mapped to four integrals (green), three of which are manifestly positive and one of which is manifestly negative.

Only \mathcal{F}_B and \mathcal{F}_D are of mixed sign. The corresponding integrals can both be resolved with Algorithm 1, at the expense of introducing algebraic transformations. Here, we demonstrate an alternative resolution procedure, first described in Ref. [199] where it was used to resolve the 1-loop box with an off-shell leg in detail. Generally, for Feynman integrals, we can perform the following changes of variables,

$$x_i \rightarrow \alpha^n x_i, \quad (4.88)$$

$$x_i \rightarrow x_j^n x_i, \quad (4.89)$$

$$x_i \rightarrow x_i + x_j \quad \& \quad x_j \rightarrow x_i + x_j, \quad (4.90)$$

where $\alpha > 0$ is a positive constant possibly depending on kinematic invariants or masses. Whenever we shift a variable such that $x_a \rightarrow x_a + x_b$, we must consider the converse case $x_b \rightarrow x_a + x_b$ to cover the entire original domain. This corresponds to bisecting the integral by inserting $\theta(x_i - x_j) + \theta(x_j - x_i) = 1$. These transformations are similar to those that appear in sector decomposition, which resolves singularities on the boundary of integration.

In Fig. 4.5, we record one possible chain of transformations for resolving integral B . The integral is mapped to four different integrals, three positive contributions and one negative contribution,

$$I_B(\mathbf{s}; \delta) = \sum_{n_+=1}^3 I_B^{+,n_+}(\mathbf{s}) + (-1 - i\delta)^{-2-3\epsilon} I_B^-(\mathbf{s}). \quad (4.91)$$

This approach generated more integrals than necessary; however, the transformations at each step are simple, as are the resulting integrands. The complete integrand of the negative contribution I_B^- , is given by,

$$I_B^-(\mathbf{s}) = \frac{1}{(s_{12} + s_{13})^{1+3\epsilon} (-s_{13})^{1+8\epsilon}} \int_{\mathbb{R}_{\geq 0}^8} \prod_{i=1}^8 dx_i \mathcal{I}_B^-(\mathbf{s}) \delta(1 - \alpha(\mathbf{x})) \quad (4.92)$$

$$\begin{aligned} \mathcal{I}_B^-(\mathbf{s}) = & x_1^{-2-3\epsilon} x_3^{-2-7\epsilon} x_8^{-1-9\epsilon} (x_2 x_4 x_6)^{-1-3\epsilon} \times \\ & \left[-s_{13} x_3 x_4 x_6 x_8 (x_7 + x_8) (x_3 + x_5 + x_7 + x_8) [s_{12} x_5 - s_{13} (x_3 + x_7 + x_8)] + \right. \\ & x_2 x_6 x_8 (x_7 + x_8) [s_{12} x_5 (x_3 + x_5) - s_{13} x_3 (x_1 + x_3 + x_5 + x_7 + x_8)] \times \\ & [s_{12} x_5 - s_{13} (x_3 + x_7 + x_8)] + x_2 x_4 x_6 (x_3 + x_5 + x_7 + x_8) \times \\ & [s_{12} x_5 (x_3 + x_5) - s_{13} x_3 (x_1 + x_3 + x_5 + x_7 + x_8)] [s_{12} x_5 - s_{13} (x_3 + x_7 + x_8)] + \\ & \left. - s_{13} x_2 x_4 x_8 (x_7 + x_8) (x_3 + x_5 + x_7 + x_8) \times \right. \\ & \left. [s_{12} x_5 (x_3 + x_5) - s_{13} x_3 (x_1 + x_3 + x_5 + x_7 + x_8)] \right]^{4\epsilon}. \end{aligned} \quad (4.93)$$

For brevity, we do not state all of the positive resolutions, they can be obtained by applying the transformations given in Fig. 4.5. Integral D is resolved in a similar way into three positive contributions and one negative contribution such that the initial integral, $I_{G..}$, can be expressed as sum over twelve integrals with manifestly positive integrands,

$$I_{G..} = 4 \left[\sum_{n_+=1}^3 I_B^{+,n_+} + \sum_{n_+=1}^3 I_D^{+,n_+} + I_E^+ + I_F^+ \right] + 4 (-1 - i\delta)^{-2-3\epsilon} [I_A^- + I_B^- + I_C^- + I_D^-]. \quad (4.94)$$

where the positive contributions I_E^+ and I_F^+ are simply I_E and I_F and the negative contributions I_A^- and I_C^- are merely I_A and I_C with $(-1 - i\delta)^{-2-3\epsilon}$ factored out accordingly.

4.3.2. Massive examples

In this section, we present 1-, 2- and 3-loop examples of integrals with massive propagators. Understanding the resolution of such integrals is important for applying the method to the calculation of massive phenomenologically-relevant amplitudes, for example, involving massive quarks, electroweak bosons or Higgs bosons. The primary complication in the massive case is that the \mathcal{F} polynomial gets modified by a term proportional to \mathcal{U} such that each Feynman parameter associated with a massive propagator may appear quadratically in the monomials of \mathcal{F} ,

$$\mathcal{F}(\mathbf{x}; \mathbf{s}) = \mathcal{F}_0(\mathbf{x}; \mathbf{s}) + \mathcal{U}(\mathbf{x}) \sum_{j=1}^N m_j^2 x_j, \quad (4.95)$$

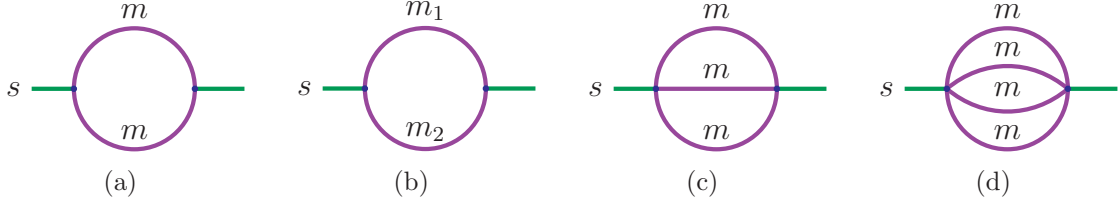


Figure 4.6.: The L -loop equal-mass bubble integrals ($L \in \{1, 2, 3\}$): the standard 1-loop bubble-, the 2-loop sunrise- and the 3-loop banana integrals.

where \mathcal{F}_0 is the polynomial corresponding to the massless version of the integral. We start by analysing the 1-loop massive bubble and triangle before applying the method to 2-loop elliptic and 3-loop hyperelliptic examples.

Equal-Mass Bubble

We saw the momentum representation of the 1-loop bubble with equal masses in Sec. 3.2.1 as an example of IBP reduction. Using the Feynman parametrisation the integral can be written as

$$J_{\text{bub}}(\mathbf{s}) = \Gamma(\epsilon) \lim_{\delta \rightarrow 0^+} I_{\text{bub}}(\mathbf{s}; \delta), \quad (4.96)$$

$$I_{\text{bub}}(\mathbf{s}; \delta) = \int_{\mathbb{R}_{\geq 0}^2} dx_1 dx_2 \frac{\mathcal{U}(\mathbf{x})^{-2+2\epsilon}}{(\mathcal{F}(\mathbf{x}; \mathbf{s}) - i\delta)^\epsilon} \delta(1 - \alpha(\mathbf{x})). \quad (4.97)$$

with the Symanzik polynomials,

$$\mathcal{U}(\mathbf{x}) = x_1 + x_2, \quad (4.98)$$

$$\mathcal{F}(\mathbf{x}; \mathbf{s}) = -sx_1x_2 + m^2(x_1 + x_2)^2. \quad (4.99)$$

Making the choice $\alpha(\mathbf{x}) = x_1 + x_2$, we can perform the x_2 integral to obtain,

$$I_{\text{bub}}(\mathbf{s}; \delta) = \int_0^1 dx_1 \frac{1}{(-sx_1(1-x_1) + m^2 - i\delta)^\epsilon}. \quad (4.100)$$

The integral develops kinematic singularities above the virtual threshold $s > 4m^2$. We consider the physical region and introduce the parametrisation

$$\beta^2 = \frac{s - 4m^2}{s} \in (0, 1), \quad (4.101)$$

After exploiting the symmetry about $x_1 = 1/2$ the integral may be written as,

$$\begin{aligned} I_{\text{bub}}(\mathbf{s}; \delta) &= \left(\frac{1 - \beta^2}{m^2} \right)^\epsilon \int_0^1 dx_1 \frac{1}{((1-x_1)^2 - \beta^2 - i\delta)^\epsilon} \\ &= \left(\frac{1 - \beta^2}{m^2} \right)^\epsilon \frac{(\beta^2)^{\frac{1}{2}-\epsilon}}{2} \tilde{I}_{\text{bub}}(\mathbf{s}; \delta), \end{aligned} \quad (4.102)$$

$$\tilde{I}_{\text{bub}}(\mathbf{s}; \delta) = \int_0^{\frac{1}{\beta^2}} dz_1 z_1^{-\frac{1}{2}} (z_1 - 1 - i\delta)^{-\epsilon}. \quad (4.103)$$

After the final change of variables we clearly see how the Minkowski regime corresponds to $z > 1$. The integral is resolved by splitting at $z = 1$ and remapping the second integral to the unit interval

$$\begin{aligned}
\tilde{I}_{\text{bub}}(\mathbf{s}; \delta) &= \int_0^1 dz_1 z_1^{-\frac{1}{2}} (z_1 - 1 - i\delta)^{-\epsilon} + \int_1^{\frac{1}{\beta^2}} dz_1 z_1^{-\frac{1}{2}} (z_1 - 1 - i\delta)^{-\epsilon}, \\
&= \int_0^1 dz_1 z_1^{-\frac{1}{2}} (z_1 - 1 - i\delta)^{-\epsilon} + \int_0^1 dz \gamma(\gamma z + 1)^{-\frac{1}{2}} (\gamma z - i\delta)^{-\epsilon}, \\
&= (-1 - i\delta)^{-\epsilon} \int_0^1 dz_1 z_1^{-\frac{1}{2}} (1 - z_1)^{-\epsilon} + \int_0^1 dz \gamma(\gamma z + 1)^{-\frac{1}{2}} (\gamma z)^{-\epsilon}, \\
&= (-1 - i\delta)^{-\epsilon} \tilde{I}_{\text{bub}}^-(\mathbf{s}) + \tilde{I}_{\text{bub}}^+(\mathbf{s}),
\end{aligned} \tag{4.104}$$

where we have introduced the abbreviation $\gamma = (1 - \beta^2)/\beta^2$ and $\gamma > 0$. With the $(-1 - i\delta)^{-\epsilon}$ factor extracted, both integrals are manifestly positive inside the domain of integration. They can be integrated numerically without contour deformation. The analytic expressions are also known in terms of a Beta function and a ${}_2F_1$ hypergeometric function respectively

$$\tilde{I}_{\text{bub}}^-(\mathbf{s}) = \int_0^1 dz_1 z_1^{-\frac{1}{2}} (1 - z_1)^{-\epsilon} = \frac{\Gamma(1 - \epsilon)}{\Gamma(\frac{3}{2} - \epsilon)} \pi^{\frac{1}{2}}, \tag{4.105}$$

$$\begin{aligned}
\tilde{I}_{\text{bub}}^+(\mathbf{s}) &= \int_0^1 dz \gamma(\gamma z + 1)^{-\frac{1}{2}} (\gamma z)^{-\epsilon} = 2 \int_0^{\sqrt{1+\gamma}-1} du u^{-\epsilon} (2 + u)^{-\epsilon} \\
&= \frac{2^{1-\epsilon}}{1 - \epsilon} (\sqrt{1 + \gamma} - 1)^{1-\epsilon} {}_2F_1 \left(1 - \epsilon, \epsilon, 2 - \epsilon, -\frac{1}{2}(\sqrt{1 + \gamma} - 1) \right).
\end{aligned} \tag{4.106}$$

Unequal-Mass Bubble

Next, we generalise the procedure for the equal-mass bubble from the previous section to the case of unequal masses, see Fig. 4.6(b). The \mathcal{F} polynomial is

$$\mathcal{F} = -s x_1 x_2 + (x_1 + x_2) (m_1^2 x_1 + m_2^2 x_2). \tag{4.107}$$

and \mathcal{U} is unchanged. We again introduce a parametrisation

$$\beta^2 = \frac{s - (m_1 + m_2)^2}{s - (m_1 - m_2)^2} \in (0, 1), \tag{4.108}$$

which reduces to Eq. (4.101) for $m_1 = m_2$. Rescaling both Feynman parameters by $x_i \rightarrow \frac{x_i}{m_i}$ yields a dimensionless polynomial to resolve

$$\tilde{\mathcal{F}} = x_1^2 + x_2^2 - 2 \frac{1 + \beta^2}{1 - \beta^2} x_1 x_2. \tag{4.109}$$

In Fig. 4.7, we plot the variety of $\tilde{\mathcal{F}}$. We find that it separates the integration domain into three regions defined by the sign of $\tilde{\mathcal{F}}$ – in regions (I) and (II) we have $\tilde{\mathcal{F}} > 0$, while in region (III) we have $\tilde{\mathcal{F}} < 0$. The goal is to split the integral into three pieces, each corresponding to one of the three regions. Solving $\tilde{\mathcal{F}} = 0$ we obtain

$$x_2 = \gamma^\pm x_1, \quad \text{with} \quad \gamma^\pm = \frac{1}{\gamma^\mp} = \frac{1 \pm \beta}{1 \mp \beta}. \tag{4.110}$$

Using Fig. 4.7 as guidance, it is easy to see how each integration domain can be mapped to the unit square with the help of Eq. (4.110). The transformations and the resulting \mathcal{F} polynomials are given by

$$\text{I :} \quad x_2 \rightarrow y_2 = x_2 + \gamma^+ x_1, \quad \tilde{\mathcal{F}}^{+,1} = x_2(x_2 + \frac{4\beta}{1-\beta^2}x_1), \quad (4.111)$$

$$\text{II :} \quad x_1 \rightarrow y_1 = x_1 + \gamma^+ x_2, \quad \tilde{\mathcal{F}}^{+,2} = x_1(x_1 + \frac{4\beta}{1-\beta^2}x_2), \quad (4.112)$$

$$\text{III :} \quad \begin{aligned} x_2 &\rightarrow y_2 = x_2 + \gamma^- x_1, \\ x_1 &\rightarrow y_1 = x_1 + \gamma^- x_2, \end{aligned} \quad \tilde{\mathcal{F}}^- = -\frac{16\beta^2}{(1-\beta)(1+\beta)^3}x_1x_2, \quad (4.113)$$

In region (I) we have mapped the $x_2 = \gamma^+ x_1$ hyperplane to $x_2 = 0$. In region (II) we have mapped the $x_2 = \gamma^- x_1$ hyperplane to $x_1 = 0$. In region (III) we have mapped the $x_2 = \gamma^- x_1$ hyperplane to $x_2 = 0$ and, simultaneously, the $x_2 = \gamma^+ x_1$ hyperplane to $x_1 = 0$. By construction, in each region, $\tilde{\mathcal{F}}$ has a fixed sign (either non-negative or non-positive).

Applying the transformations also to the \mathcal{U} polynomial and including the Jacobian, the original integral is expressed as a sum of three integrals over manifestly positive integrands

$$I_{\text{bub}, m_1 \neq m_2} = I_{\text{bub}, m_1 \neq m_2}^{+,1} + I_{\text{bub}, m_1 \neq m_2}^{+,2} + (-1 - i\delta)^{-\epsilon} I_{\text{bub}, m_1 \neq m_2}^-, \quad (4.114)$$

where

$$I_{\text{bub}, m_1 \neq m_2}^{+,1} = (m_1 m_2)^{1-2\epsilon} (1-\beta)^{2-2\epsilon} \int_{\mathbb{R}_{\geq 0}^2} dx_1 dx_2 x_2^{-\epsilon} \left(x_2 + \frac{4\beta}{1-\beta^2} x_1 \right)^{-\epsilon} \times \\ (m_1(1-\beta)x_2 + [m_1(1+\beta) + m_2(1-\beta)]x_1)^{-2+2\epsilon} \delta(1-\alpha(\mathbf{x})), \quad (4.115)$$

$$I_{\text{bub}, m_1 \neq m_2}^{+,2} = (m_1 m_2)^{1-2\epsilon} (1-\beta)^{2-2\epsilon} \int_{\mathbb{R}_{\geq 0}^2} dx_1 dx_2 x_1^{-\epsilon} \left(x_1 + \frac{4\beta}{1-\beta^2} x_2 \right)^{-\epsilon} \times \\ (m_2(1-\beta)x_1 + [m_2(1+\beta) + m_1(1-\beta)]x_2)^{-2+2\epsilon} \delta(1-\alpha(\mathbf{x})), \quad (4.116)$$

$$I_{\text{bub}, m_1 \neq m_2}^- = (4m_1 m_2 \beta)^{1-2\epsilon} (1-\beta^2)^\epsilon \int_{\mathbb{R}_{\geq 0}^2} dx_1 dx_2 (x_1 x_2)^{-\epsilon} \times \\ [(m_1 + m_2)(x_1 + x_2) - (m_1 - m_2)(x_1 - x_2)\beta]^{-2+2\epsilon} \delta(1-\alpha(\mathbf{x})). \quad (4.117)$$

Each integral can be evaluated numerically without having to use contour deformation.

1-loop Triangle with an Off-Shell Leg

For our final 1-loop massive example, we consider the massive triangle with an off-shell leg (the independent equal-mass configurations of which are shown in Fig. 4.8). We present the resolution of the fully massive triangle in detail as it is the most complicated configuration. The other ones may be resolved using a similar procedure. The integral we consider is

$$J_{\text{tri}} = \lim_{\delta \rightarrow 0^+} -\Gamma(1+\epsilon) I_{\text{tri}} \quad (4.118)$$

$$I_{\text{tri}} = \int_{\mathbb{R}_{\geq 0}^3} dx_1 dx_2 dx_3 \frac{(x_1 + x_2 + x_3)^{-1+2\epsilon}}{(-p^2 x_1 x_2 + m^2 (x_1 + x_2 + x_3)^2 - i\delta)^{1+\epsilon}} \delta(1-\alpha(\mathbf{x})). \quad (4.119)$$

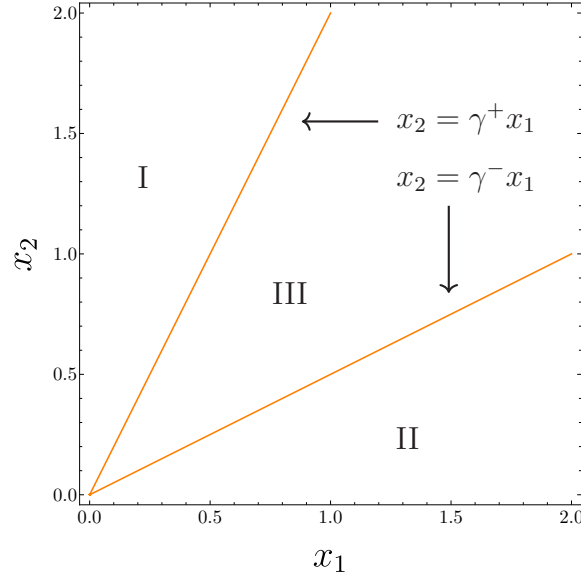


Figure 4.7.: The variety of $\tilde{\mathcal{F}}$ and the three regions of the integration domain which it separates. In regions (I) and (II), $\tilde{\mathcal{F}} > 0$ whereas in region (III), $\tilde{\mathcal{F}} < 0$. The kinematic regime is given by $\beta^2 \in (0, 1)$ (with $m_1^2, m_2^2 > 0$) equivalent to $\gamma^+ = \frac{1}{\gamma^-} > 0$.

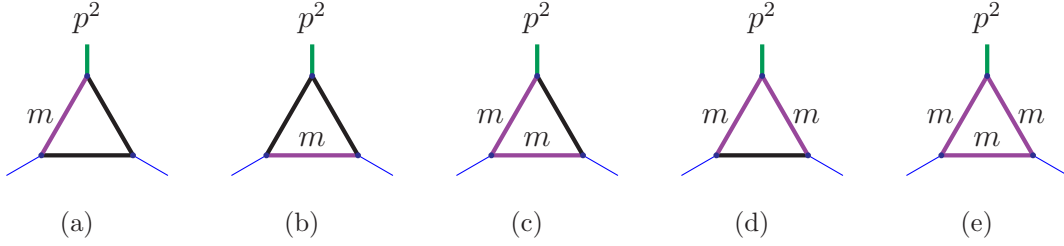


Figure 4.8.: Independent equal mass triangles with an off-shell leg ($p^2 > 0$)

It is possible to make multiple choices to parametrise the projective integral. Making different choices of hyperplane $\alpha(\mathbf{x}) = \sum_i \alpha_i x_i$ can lead to different solutions. Experience shows that using the symmetry of the problem, where possible, tends to produce the neatest solutions. Breaking the symmetry with a choice like $\delta(1 - x_1)$ would still work but often complicates the integrands, for example by introducing square roots of the Feynman parameters. Following this philosophy, we make the symmetric choice $\delta(1 - x_1 - x_2 - x_3)$. This is often the optimal choice at 1-loop as it sets $\mathcal{U} = 1$.

Defining $\beta^2 = \frac{p^2 - 4m^2}{p^2} \in (0, 1)$ and using the δ -function to perform the integral over x_3 , we have

$$I_{\text{tri}} = \left(\frac{1 - \beta^2}{m^2} \right)^{1+\epsilon} \tilde{I}_{\text{tri}} \quad (4.120)$$

$$\tilde{I}_{\text{tri}} = \int_{\mathbb{R}_{\geq 0}^2} dx_1 dx_2 \theta(1 - x_1 - x_2) (1 - \beta^2 - 4x_1 x_2 - i\delta)^{-1-\epsilon} \quad (4.121)$$

where the Heaviside function is a result of the symmetric choice of hyperplane and restricts the remaining domain of integration to a simplex. The transformation $x_2 \rightarrow (1 - x_1)x_2$ maps the integration domain to the unit square. The resulting integral is

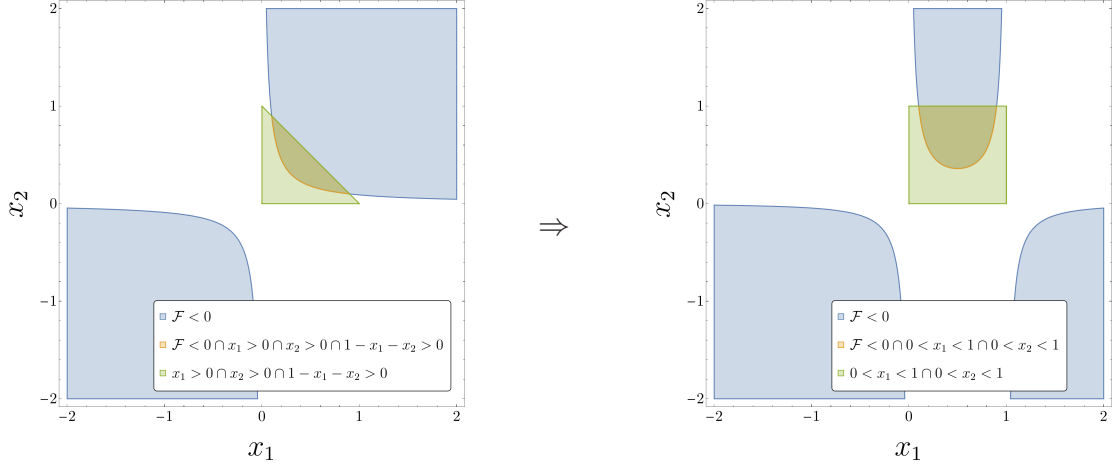


Figure 4.9.: Remapping the simplex integration region of the massive triangle (in green) to the positive unit square in $\mathbb{R}_{\geq 0}^2$. Here, \mathcal{F} is to be understood as \mathcal{F} after the δ -function has been integrated out and in the second panel, after the remapping transformation. The kinematic regime is given by $\beta^2 \in (0, 1)$ (with $m^2 > 0$).

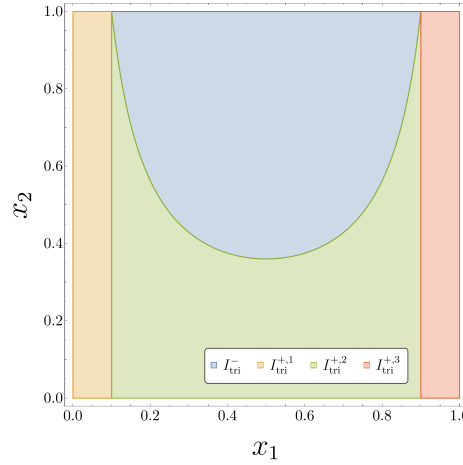


Figure 4.10.: The integration domain of the massive triangle separated into one negative and three positive regions.

$$\tilde{I}_{\text{tri}} = \int_0^1 dx_1 dx_2 (1 - x_1) (1 - \beta^2 - 4(1 - x_1)x_1x_2 - i\delta)^{-1-\epsilon}. \quad (4.122)$$

To understand how to split Eq. (4.122) into a sum of integrals of positive integrands, we consider again the variety of \mathcal{F} . Visualising where $\mathcal{F} = 0$ intersects the boundary of the unit square inspires the decomposition seen in Fig. 4.10. We have three regions (red, green, yellow) where $\mathcal{F} > 0$ and one region (blue) where $\mathcal{F} < 0$.

The goal is now to map each region to the full unit square. For brevity, we explicitly show the steps for the negative region and state the results for the remaining positive regions. Firstly, we need to map the orange and red regions outside the square. To do this, we find the x_1 -values where $\mathcal{F} = 0$ intersects the boundary of the domain at $x_2 = 1$. A trivial calculation reveals these values to be $\frac{1 \pm \beta}{2}$ and we want to map the

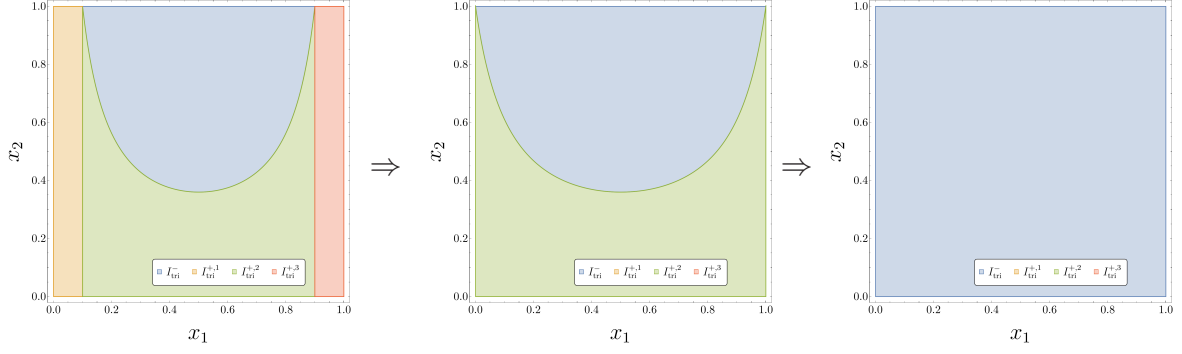


Figure 4.11.: The chain of transformations which maps the negative region of the massive triangle (in blue) to the positive unit square.

x_1 -lines defined by these values to the boundaries of integration $x_1 = 0$ and $x_1 = 1$. We can easily construct a transformation which satisfies these demands:

$$x'_1 \stackrel{!}{=} \frac{x_1 - \frac{1-\beta}{2}}{\frac{1+\beta}{2} - \frac{1-\beta}{2}} \quad \Rightarrow \quad x_1 \rightarrow \frac{1}{2} + \beta \left(x_1 - \frac{1}{2} \right) \quad (4.123)$$

and we show the effect of this transformation in the transition from the first to the second panel of Fig. 4.11 with $\mathcal{F} \rightarrow 1 - \beta^2 - (1 - (1 - 2x_1)^2 \beta^2) x_2$. The next step is to map the variety of the transformed \mathcal{F} to the boundary at $x_2 = 0$. Solving for x_2 we find

$$x_2 = \frac{1 - \beta^2}{1 - (1 - 2x_1)^2 \beta^2} = f(x_1) \quad (4.124)$$

and the required transformation is thus given by

$$x'_1 \stackrel{!}{=} x_1, \quad x'_2 \stackrel{!}{=} \frac{x_2 - f(x_1)}{1 - f(x_1)} \quad \Rightarrow \quad x_1 \rightarrow x_1, \quad x_2 \rightarrow x_2 + (1 - x_2) f(x_1). \quad (4.125)$$

This transformation generates the transition between the second and third panels in Fig 4.11 and completes the mapping of the negative region to the positive unit square as required. The final result for the negative contribution after taking into account the Jacobian determinants of the transformations is

$$\tilde{I}_{\text{tri}}^- = 2^{-1-2\epsilon} \beta^{1-2\epsilon} \int_0^1 dx_1 dx_2 (1 - x_1)^{-\epsilon} x_1^{-\epsilon} x_2^{-1-\epsilon} (1 - (1 - 2x_1) \beta)^{-1} \quad (4.126)$$

where we have already factored out $(-1 - i\delta)^{-1-\epsilon}$. We note that this integral is $\mathcal{O}(\frac{1}{\epsilon})$ whereas the full integral I_{tri} is finite. This pole cancels exactly with a corresponding pole in $I_{\text{tri}}^{+,2}$ and, since $I_{\text{tri}}^{+,1}$ and $I_{\text{tri}}^{+,3}$ are free of poles in ϵ , the construction consistently reproduces the full result.

It is illuminating to consider the general structure of this cancellation; if the full result is finite and also has an imaginary part at the leading order (which is true of I_{tri}), the negative contribution must necessarily have a pole in ϵ to generate the imaginary part from multiplying the ϵ^1 term in the expansion of $(-1 - i\delta)^{a+b\epsilon}$. Furthermore, the total positive contribution must have an equal pole in ϵ to cancel the pole of the negative contribution leaving a finite result. This type of analysis can be fruitful in predicting a priori the pole structure of the constituent integrals in the decomposition.

For completeness, we state the result for the positive contributions:

$$\tilde{I}_{\text{tri}}^{+,1} = \frac{1}{4} (1 - \beta)^{-\epsilon} \int_0^1 dx_1 dx_2 (2 - (1 - \beta) x_1) (1 + \beta - (2 - (1 - \beta) x_1) x_1 x_2)^{-1-\epsilon} \quad (4.127)$$

$$\tilde{I}_{\text{tri}}^{+,2} = \frac{\beta}{2} (1 - \beta^2)^{-\epsilon} \int_0^1 dx_1 dx_2 (1 - x_2)^{-1-\epsilon} (1 - (1 - 2x_1) \beta)^{-1} \quad (4.128)$$

$$\tilde{I}_{\text{tri}}^{+,3} = \frac{1}{4} (1 - \beta)^{1-\epsilon} \int_0^1 dx_1 dx_2 (1 - x_1) (1 + \beta - x_2 (1 - x_1) (1 + \beta + (1 - \beta) x_1))^{-1-\epsilon}. \quad (4.129)$$

This gives the total result

$$I_{\text{tri}} = \sum_{n_+=1}^3 I_{\text{tri}}^{+,n_+} + (-1 - i\delta)^{-1-\epsilon} I_{\text{tri}}^-, \quad (4.130)$$

where we stress again that each integrand in the constituent integrals of Eq. (4.130) is manifestly positive in the Minkowski region defined by $\beta \in (0, 1)$ throughout the entire integration domain.

2-loop Elliptic Sunrise

In this section, we analyse the equal mass sunrise integral, see Fig. 3.1(b), the simplest integral which involves a function class beyond polylogarithms. The sunrise has been studied extensively, see e.g. [200–203], and is known to evaluate to elliptic integrals. Using the sunrise integral as an example, we demonstrate that there is no fundamental obstruction to avoiding contour deformation for massive integrals beyond polylogarithms. Contrary to previous examples, however, we find that the sunrise leads us naturally to algebraic transformations of the Feynman parameters instead of purely rational function transformations. To the best of our current knowledge, it seems not to be possible to find a resolution which avoids square roots in the transformations; however, this does not present a fundamental obstacle for numerical evaluation, as demonstrated in Sec. 4.4.

Using the Feynman parametrisation, the sunrise integral can be written as

$$J_{\text{sun}} = \lim_{\delta \rightarrow 0^+} -\Gamma(-1 + 2\epsilon) I_{\text{sun}} \quad (4.131)$$

$$I_{\text{sun}} = \int_{\mathbb{R}_{\geq 0}^3} dx_1 dx_2 dx_3 \frac{(x_1 x_2 + x_2 x_3 + x_1 x_3)^{-3+3\epsilon} \delta(1 - \alpha(\mathbf{x}))}{(-s x_1 x_2 x_3 + (x_1 x_2 + x_2 x_3 + x_1 x_3) m^2 (x_1 + x_2 + x_3) - i\delta)^{-1+2\epsilon}}. \quad (4.132)$$

The virtual threshold is $s > 9m^2$ and we use the parametrisation $\beta^2 = \frac{s-9m^2}{s} \in (0, 1)$. For the Dirac-delta constraint, we again use the symmetry with the choice $\delta(1 - x_1 - x_2 - x_3)$ and integrate out x_3 which results in the integral

$$I_{\text{sun}} = \left(\frac{\tilde{\beta}}{m^2} \right)^{-1+2\epsilon} \int_0^1 dx_1 dx_2 \frac{(1 - x_1)^{-1+\epsilon} (x_1 + (1 - x_1) (1 - x_2) x_2)^{-3+3\epsilon}}{\left((\tilde{\beta} - 9x_1) (1 - x_1) (1 - x_2) x_2 + \tilde{\beta} x_1 - i\delta \right)^{-1+2\epsilon}}, \quad (4.133)$$

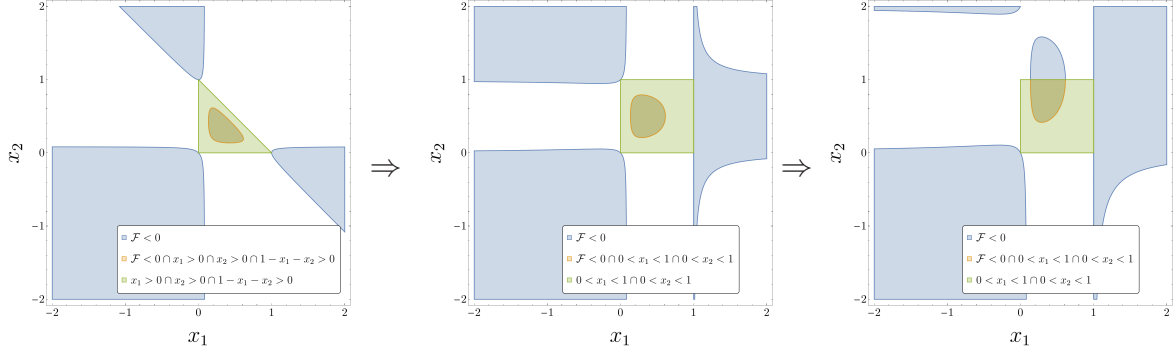


Figure 4.12.: Remapping the simplex integration region of the elliptic sunrise (in green) to the positive unit square in $\mathbb{R}_{\geq 0}^2$ then exploiting the symmetry about $x_2 = \frac{1}{2}$. Here, \mathcal{F} is to be understood as \mathcal{F} after the δ -function has been integrated out and in the second and third panels, after their respective remapping transformations as well.

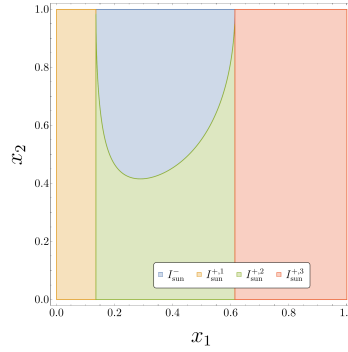


Figure 4.13.: The integration domain of the elliptic sunrise separated into one negative and three positive regions.

with $\tilde{\beta} = 1 - \beta^2$. Eq. (4.133) reveals that the integrand has a symmetry under $x_2 \rightarrow 1 - x_2$. This allows integrating x_2 from 0 to $\frac{1}{2}$ instead, and then double the result. We remap this halved integration domain back to the positive unit square (as shown in the transition between the second and third panels of Fig. 4.12) and benefit from exploiting this symmetry which reduces the number of regions we will need to resolve. Were this symmetry not present, it would still be possible to resolve the integral, but with more regions.

$$I_{\text{sun}} = 4^{2-\epsilon} \left(\frac{\tilde{\beta}}{m^2} \right)^{-1+2\epsilon} \int_0^1 dx_1 dx_2 \frac{(1-x_1)^{-1+\epsilon} (4x_1 + (1-x_1)(2-x_2)x_2)^{-3+3\epsilon}}{\left((\tilde{\beta} - 9x_1)(1-x_1)(2-x_2)x_2 + 4\tilde{\beta}x_1 - i\delta \right)^{-1+2\epsilon}}. \quad (4.134)$$

We proceed in the same way as for the massive triangle from the previous section. We define three positive regions (red, green, yellow), where the denominator of Eq. (4.134), which we will loosely refer to as \mathcal{F} , is positive and one negative region (blue) where it is negative. The decomposition can be seen in Fig. 4.13. We will resolve the negative region in detail once more as it plays the special role in solely generating the imaginary part of the full integral. The positive regions can be resolved in very similar manners.

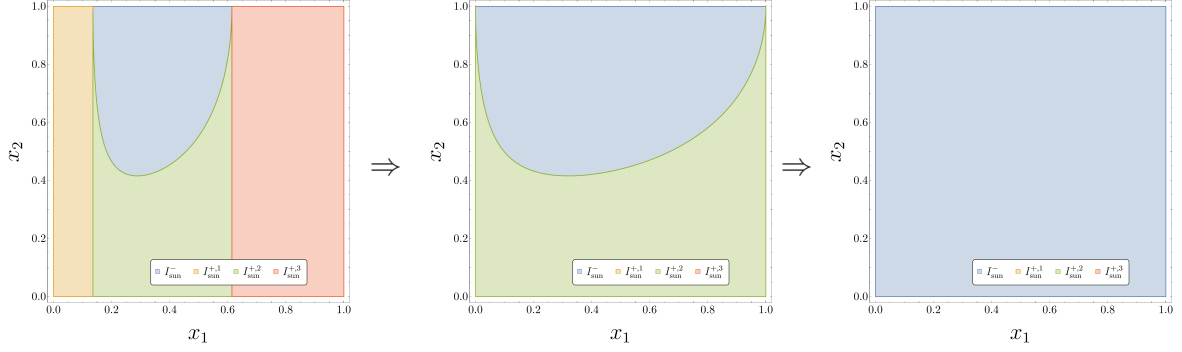


Figure 4.14.: The chain of transformations which maps the negative region of the elliptic sunrise (in blue) to the positive unit square.

To resolve the negative region, we first must map the two positive rectangles to the boundary of the square. Using the intersection of $\mathcal{F} = 0$ with the boundary at $x_2 = 1$, we find the associated x_1 values and construct the transformation

$$\begin{aligned}
 x'_1 &\stackrel{!}{=} \frac{x_1 - \frac{1}{6}(2 + \beta^2 - \beta\bar{\beta})}{\frac{1}{6}(2 + \beta^2 + \beta\bar{\beta}) - \frac{1}{6}(2 + \beta^2 - \beta\bar{\beta})} \\
 &\Downarrow \\
 x_1 &\rightarrow \frac{1}{6}(2 + \beta^2 - (1 - 2x_1)\beta\bar{\beta}), \tag{4.135}
 \end{aligned}$$

with $\bar{\beta} = \sqrt{8 + \beta^2}$. The effect of this transformation is shown in the transition from the first to the second panel in Fig. 4.14. It is clear from this geometric picture what the next (and final) step of the resolution should be: mapping the variety to $x_2 = 0$ keeping the boundary at $x_2 = 1$ fixed. The main distinction between this case and the massive triangle enters here; we want to solve the variety as $x_2 = f(x_1)$ but this gives us a function involving a square root containing x_1 instead of a purely rational function. We find that, picking the relevant square root solution for the transformed \mathcal{F} ,

$$\begin{aligned}
 \mathcal{F} &= 0 \\
 &\Downarrow \\
 x_2 &= 1 - \frac{3(1 - x_1)x_1\beta\bar{\beta}^2}{\sqrt{3(1 - x_1)x_1\bar{\beta}^2(4 - \beta^2(2(1 + \beta^2) - 3(1 - x_1)x_1\bar{\beta}^2) - 2(1 - 2x_1)\beta\tilde{\beta}\bar{\beta})}} = f(x_1). \tag{4.136}
 \end{aligned}$$

Applying the same set of transformations to the \mathcal{U} polynomial and accounting for the corresponding Jacobian determinants, we obtain for our final result for the negative contribution,

$$\begin{aligned}
 I_{\text{sun}}^- &= 2^{7-6\epsilon} 3^{\frac{1}{2}-\epsilon} (\beta^2)^{2-2\epsilon} (\bar{\beta}^2)^{2-2\epsilon} \left(\frac{\tilde{\beta}}{m^2} \right)^{-1+2\epsilon} \\
 &\times \int_0^1 dx_1 dx_2 (1 - x_1)^{\frac{3}{2}-2\epsilon} x_1^{\frac{3}{2}-2\epsilon} x_2^{1-2\epsilon} R_{\text{sun}}^-(x_1, x_2; \beta), \tag{4.137}
 \end{aligned}$$

where the finite remainder function is given by,

$$R_{\text{sun}}^-(x_1, x_2; \beta) = R_1(x_2; \beta) R_2(x_1; \beta) R_3(x_1; \beta) R_4(x_1, \beta) R_5(x_1, x_2; \beta), \quad (4.138)$$

$$R_1(x_2; \beta) = \bar{x}_2^{1-2\epsilon}, \quad (4.139)$$

$$R_2(x_1; \beta) = [-\beta^2 + \beta\bar{\beta}\tilde{x}_1 + 4]^{3\epsilon-2}, \quad (4.140)$$

$$R_3(x_1; \beta) = \left[4 - \beta \left(2\beta(\beta^2 + 1) - 3\beta\bar{\beta}^2 x_1 \bar{x}_1 + 2\tilde{\beta}\bar{\beta}\tilde{x}_1 \right) \right]^{\frac{3}{2}-\epsilon}, \quad (4.141)$$

$$R_4(x_1; \beta) = \left[\beta^2 \bar{\beta}^2 x_1 \bar{x}_1 (-11\beta^2 + 3\beta\bar{\beta}\tilde{x}_1 + 20) + 4\tilde{\beta}^2 (\beta^2 - \beta\bar{\beta}\tilde{x}_1 + 4) \right]^{1-2\epsilon}, \quad (4.142)$$

$$R_5(x_1, x_2; \beta) = \left[\beta^2 \bar{\beta}^2 x_1 \bar{x}_1 (x_2 \bar{x}_2 (-\beta^2 + \beta\bar{\beta}\tilde{x}_1 + 4) + 4\beta(3\beta - \bar{\beta}\tilde{x}_1)) + 4\tilde{\beta}(\beta^4 + 7\beta^2 - (\beta^2 + 3)\beta\bar{\beta}\tilde{x}_1 + 4) \right]^{3\epsilon-3}, \quad (4.143)$$

with $\bar{x}_1 = 1 - x_1$, $\tilde{x}_1 = 1 - 2x_1$, $\bar{x}_2 = 2 - x_2$ and $\bar{\beta} = \sqrt{8 + \beta^2}$, $\tilde{\beta} = 1 - \beta^2$. Each of the factors in the integrand of (4.137) can be shown to be positive for $0 < x_1 < 1$, $0 < x_2 < 1$ and $0 < \beta < 1$, thereby removing the need for a contour deformation.

We can perform similar resolutions for the positive regions to obtain the overall construction:

$$I_{\text{sun}} = \sum_{n_+=1}^3 I_{\text{sun}}^{+,n_+} + (-1 - i\delta)^{1-2\epsilon} I_{\text{sun}}^-, \quad (4.144)$$

which resolves the elliptic sunrise and removes the need for contour deformation upon numerical integration.

3-loop Hyperelliptic Banana

The final example is the equal-mass three-loop banana integral. The underlying geometry is a Calabi-Yau 2-fold ($K3$ surface) and the banana is sometimes referred to as being hyperelliptic. We make no further comment on the underlying geometry associated to these integrals except to say that this motivates studying these particular examples by proving that these geometric properties are not inherently prohibitive to the resolution procedure.

Using the Feynman parametrisation the three-loop banana integral can be written as

$$J_{\text{ban}} = \lim_{\delta \rightarrow 0^+} \Gamma(-2 + 3\epsilon) I_{\text{ban}} \quad (4.145)$$

$$I_{\text{ban}} = \int_{\mathbb{R}_{\geq 0}^4} \prod_{i=1}^4 dx_i \frac{(x_1 x_2 x_3 + x_1 x_3 x_4 + x_2 x_3 x_4 + x_1 x_2 x_4)^{-4+4\epsilon} \delta(1 - \alpha(\mathbf{x}))}{(\mathcal{F}(\mathbf{x}, \mathbf{s}) - i\delta)^{2-3\epsilon}} \quad (4.146)$$

where the \mathcal{F} polynomial is

$$\mathcal{F}(\mathbf{x}, \mathbf{s}) = -sx_1 x_2 x_3 x_4 + (x_1 x_2 x_3 + x_1 x_3 x_4 + x_2 x_3 x_4 + x_1 x_2 x_4) m^2 (x_1 + x_2 + x_3 + x_4). \quad (4.147)$$

We proceed in mostly the same way as for the massive triangle and sunrise integrals. The Minkowski regime is parametrised by $\beta^2 = \frac{s-16m^2}{s} \in (0, 1)$. We again use a symmetric choice of hyperplane and integrate out x_4 . The resulting simplex integration domain is then remapped to the positive unit cube. The symmetry of the resulting

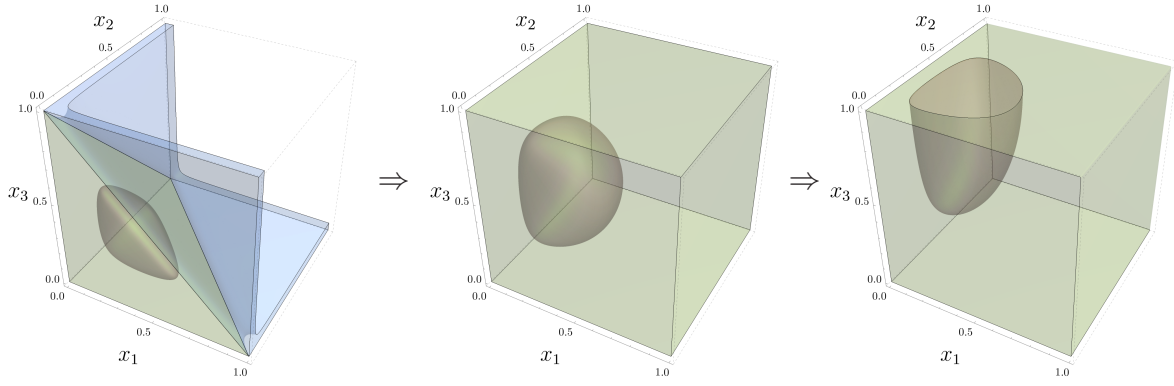


Figure 4.15.: Remapping the simplex integration region of the banana (in green) to the positive unit cube in $\mathbb{R}_{\geq 0}^3$ then exploiting the symmetry about $x_3 = \frac{1}{2}$. We omit the legend for clarity but this figure should be understood analogously to Fig. 4.12 with the $\mathcal{F} < 0$ region given in blue outside the domain of integration and in orange within. Here, \mathcal{F} is to be understood as \mathcal{F} after the δ -function has been integrated out and in the second and third panels, after their respective remapping transformations as well. The kinematic regime is given by $\beta^2 \in (0, 1)$ (with $m^2 > 0$).

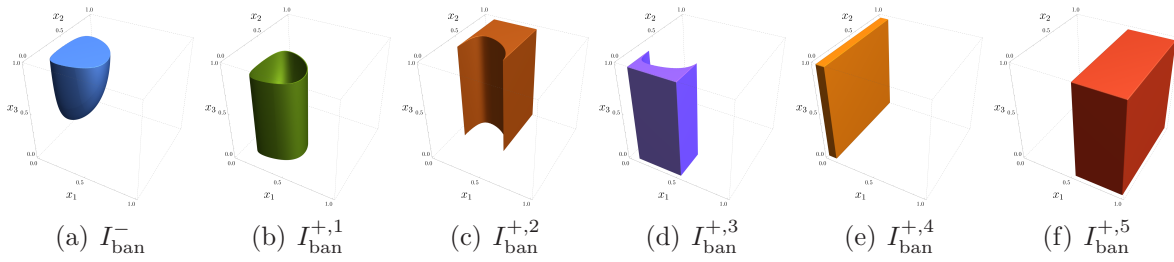


Figure 4.16.: The six regions (one negative, 4.16(a), and five positive, 4.16(b) – 4.16(f)) into which the integration domain of the banana is partitioned in this resolution.

integrand under $x_3 \rightarrow 1 - x_3$ allows us once again to reduce the number of integrals into which we will eventually decompose I_{ban} by integrating instead from $x_3 = 0$ up to $x_3 = \frac{1}{2}$. This sequence of domain-remapping transformations is depicted in Fig. 4.15 (where this time we focus solely on the positive unit cube) and should be compared and contrasted with the corresponding set of transformations for the elliptic sunrise in Fig. 4.12.

We now repeat the same decomposition procedure as for the triangle and sunrise, being guided by the visual geometry of $\mathcal{F} = 0$. The art of partitioning the integration domain revolves around the balance between minimising the number of integrals in the decomposition (minimally one negative region and one positive region) and producing regions which can all be successfully mapped back to the positive unit hypercube. For this example, we have selected six regions, see Fig. 4.16, which generates five positive contributions and one negative contribution.

The remapping to the unit cube works very similarly to the resolution of the sunrise in the previous section. We therefore leave out the explicit steps and instead sketch the procedure for the negative piece schematically. First, we look at the intersection of the variety of the transformed \mathcal{F} polynomial with the plane $x_3 = 1$ and analyse the resulting closed curve. It is clear from focusing on the plane $x_3 = 1$ in the first panel

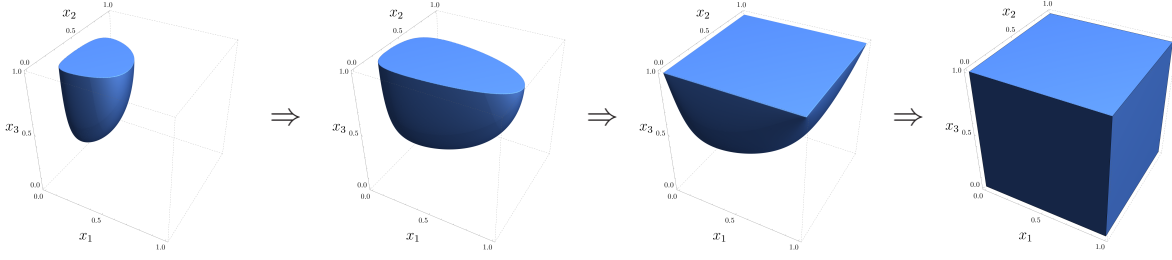


Figure 4.17.: The chain of transformations which maps the negative region of the banana (in blue) to the positive unit cube.

of Fig. 4.17 that this curve has four turning points (by which we mean the four points on the curve of maximal and minimal x_1 and x_2 respectively). In the resolution of the negative region shown in Fig. 4.17, we arbitrarily choose to first identify the points of maximal and minimal x_1 on the curve using standard turning point analysis and then map the planes $x_1 = x_1^{\min}$ and $x_1 = x_1^{\max}$ to $x_1 = 0$ and $x_1 = 1$ respectively. This transformation is shown in the transition between the first and second panels of Fig. 4.17. On the curve, both x_1^{\min} and x_1^{\max} have a corresponding x_2 -value of $x_2 = \frac{1}{3}$ (independent of β) and taking a slice through the integration domain at this value of x_2 (combining all regions in Fig. 4.16) generates a plot which is, superficially, very similar to Fig. 4.13. The next step is to solve the transformed curve at $x_3 = 1$ as $x_2 = f(x_1)$. Clearly, from the second panel of Fig. 4.17, this will have two solutions (the curve is quadratic in x_2) and a square root involving the Feynman parameters – specifically, x_1 – enters at this point. The map which takes $x_2 = f_1(x_1)$ to the plane $x_2 = 1$ and $x_2 = f_2(x_1)$ to the plane $x_2 = 0$ is trivial to construct once f_1 and f_2 are known and we demonstrate the effect of this transformation in the transition between the second and third panels of Fig. 4.17. Finally, we have to solve the variety of the transformed \mathcal{F} as $x_3 = g(x_1, x_2)$ (which also has two solutions due to the quadratic appearance of x_3 but only one solution which is relevant within the integration domain) and map this to the plane $x_3 = 0$ while keeping $x_3 = 1$ fixed. This transformation is shown in the transition between the third and fourth panels of Fig. 4.17 and concludes the mapping of the original negative region to the positive unit cube.

A similar analysis is carried out for the remaining five positive regions in Fig. 4.16, resulting in²⁵

$$I_{\text{ban}} = \sum_{n_+=1}^5 I_{\text{ban}}^{+,n_+} + (-1 - i\delta)^{1-2\epsilon} I_{\text{ban}}^- \quad (4.148)$$

where all of the integrands appearing in Eq. (4.148) are manifestly positive within the domain of integration (and away from the boundary).

4.4. Benchmarks

In this section we numerically evaluate the resolved integrals from Sec. 4.3.1 and Sec. 4.3.2 with PYSECDEC v1.6.4 and compare the timings to when using contour deformation. We emphasise that PYSECDEC is not optimised for this new approach

²⁵For the numerical evaluation in PYSECDEC, a manual sector decomposition of one of the positive contributions into two constituent integrals is performed. This is implementation-dependent and orthogonal to the resolution procedure for avoiding contour deformation.

and its efficiency could be greatly improved for resolved integrals. For example, parsing prefactor expressions can dominate the evaluation time, primarily due to the inefficient `SymPy` routines used. In contrast, the same expressions load almost instantly in systems like `Mathematica`. These expressions appear as a result of the resolution procedure, particularly in the massive examples, and have therefore not been a bottleneck in `PYSECDEC` before. We subtract the loading time of all prefactors from the results, and perform our comparisons strictly on the actual *integration time*. This issue is particularly relevant in the case of the 1-loop massive triangle. Additional inefficiencies in the current implementation of `PYSECDEC` include the re-evaluation of common sub-expressions appearing across multiple ϵ orders and the unnecessary expansion of finite expressions (especially those appearing at higher orders in ϵ in finite integrals). Parts of the current implementation also unnecessarily utilise complex numbers, even for the real integrands we obtain after resolution. For this reason, the speed-up factors presented in this section should be interpreted as a pessimistic estimate of the impact of the resolution procedure.

Unless otherwise specified, the integration was performed using the `Disteval` integrator, which was run on an NVIDIA A100 80G GPU. The `PYSECDEC` integration libraries were compiled²⁶ using `CUDA 12.4.131`. The exceptions are the 2L sunrise and 3L banana integrals, where an older integrator was used in order to access the feature of user-defined `C++` functions. The resolved versions of these examples contain large positive remainder functions raised to high integer powers, that by default get expanded into very large expressions by `FORM` routines within `PYSECDEC`. Future versions of `PYSECDEC` should therefore offer the option to prevent certain functions from being expanded. For now, this problem can be circumvented by manually defining the remainder expressions as symbolic functions, and providing them directly as `C++` functions. The integration libraries for these two examples were compiled²⁷ with `gcc 7.5.0` and run on one core of an AMD EPYC 7352 CPU.

We now proceed by presenting benchmarks for each resolved integral, comparing integration times with and without contour deformation. Both massless and massive integrals are evaluated for increasing centre-of-mass energies with other kinematics fixed. Additionally, the massive integrals are benchmarked in the small-mass regime, where the centre-of-mass energy is fixed and the internal mass m is pushed to very small values. We emphasise that for the massive examples shown here, results are not obtained using Algorithm 1 and instead various possible generalisations of this algorithm were explored. In particular, results are obtained after integrating out the Dirac-delta functional with a particular choice of hyperplane. This is not necessarily the optimal choice for numerical performance. The goal in Sec. 4.3.2 was to show that avoiding contour deformation for massive integrals is in principle possible. Moreover, the massive examples are all integrals with few internal lines. As described in the introduction to this chapter, the performance gain from avoiding contour deformation is expected to be the largest for more complicated integrals with many internal lines.

²⁶Using the following make command: `make disteval SECDEC_WITH_CUDA_FLAGS=-arch=sm_80 CXXFLAGS="-O3 -mfma -mavx2" CXX=g++-12.`

²⁷Using the simple make command: `make pylink.`

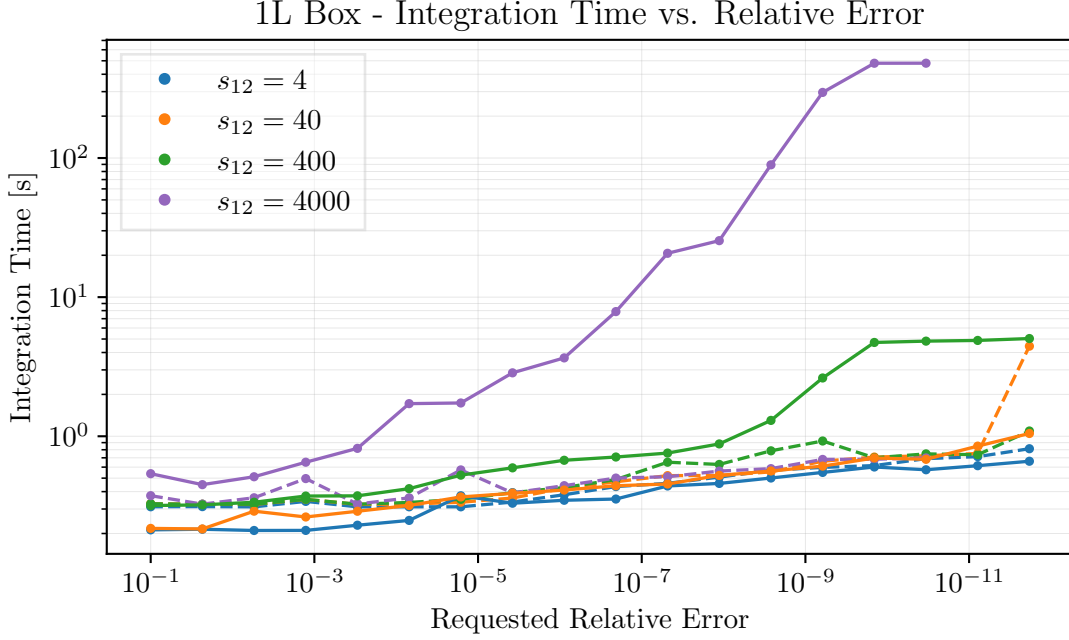


Figure 4.18.: Timings with (solid lines) and without (dashed lines) contour deformation for the massless 1-loop box with off-shell legs, expanded up to the finite order. Evaluated for different values of s_{12} with $s_{13} = -1$ and $p_1^2 = 2$ fixed.

4.4.1. 1-loop box with an off-shell leg

Figure 4.18 shows the integration times for the 1-loop box with one off-shell leg, with and without contour deformation, for increasing values of the centre-of-mass energy s_{12} , keeping $s_{13} = -1$ and $p_1^2 = 2$ fixed. For non-extreme values of s_{12} , we do not observe significant differences between the two versions, which is expected since the 1-loop box is a very simple integral and the time is dominated by fixed overheads, e.g. generating the QMC sampling lattice. However, the timings for large s_{12} reveal a general feature that persists in more complicated integrals: the relative performance of the contour deformation is much worse in extreme kinematic configurations. For example, to obtain 10 digits of precision with $s_{12} = 4000$, avoiding contour deformation yields a factor of $\frac{480s}{0.7s} \approx 686 \times$ speed-up.

4.4.2. 1-loop on-shell pentagon

Figure 4.19 shows the integration times for the 1-loop pentagon, with and without contour deformation, for increasing values of the centre-of-mass energy s_{12} , keeping the other kinematics fixed as $(s_{23}, s_{34}, s_{45}, s_{51}) = (-3, 2.5, -3, 5)$. We see similar behaviour to the 1-loop box; the performance differences are more significant for large s_{12} . In this case, however, in addition to the aforementioned behaviour being more extreme, the resolved integrands reach high precision more quickly even for non-extreme s_{12} values. For example, to obtain 7 digits of precision with $s_{12} = 400$, avoiding contour deformation yields a factor of $\frac{2021}{2.21} \approx 914 \times$ speed-up. For the same number of digits with $s_{12} = 4$, we instead get a factor of $\frac{11.5}{2.15} \approx 5.3 \times$ speed-up. For $s_{12} \geq 4000$, it is difficult to obtain any digits at all with contour deformation, while the resolved integral is essentially insensitive to this increase.

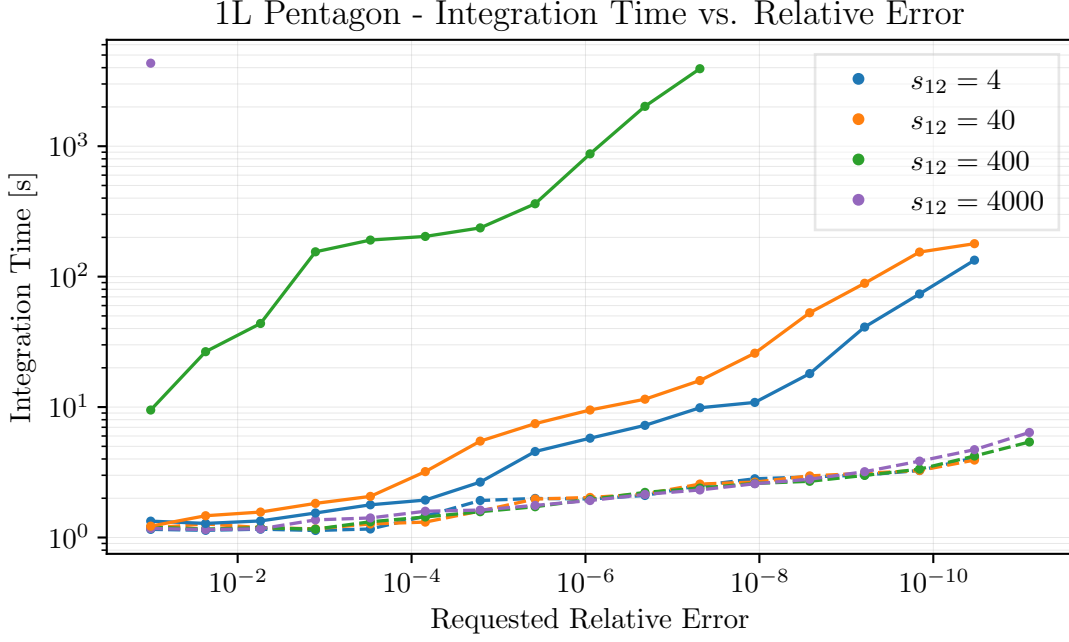


Figure 4.19.: Timings with (solid lines) and without (dashed lines) contour deformation for the massless 1-loop pentagon, expanded up to the finite order. Evaluated for different values of s_{12} while the other kinematics are fixed at $(s_{23}, s_{34}, s_{45}, s_{51}) = (-3, 2.5, -3, 5)$.

4.4.3. 2-loop non-planar boxes

Figures 4.20 and 4.21 show the integration times for the 2-loop non-planar boxes with 6 and 7 propagators respectively (BNP6 and BNP7), with and without contour deformation, for increasing values of the centre-of-mass energy s_{12} , keeping $s_{23} = -1$ fixed. These integrals have quite a similar structure, but BNP7 is much more challenging numerically. In fact, for BNP7, numerical integration with contour deformation failed to yield any digits of precision across all tested kinematic configurations, within a 10-hour time limit. The resolved BNP7, however, converges with reasonable precision across the entire range in s_{12} , though the integration time does deteriorate for large values of s_{12} . The results for the resolved integral have therefore instead been checked and validated against the analytic expression from Ref. [204], in which both integrals were computed. The integration with contour deformation eventually converged to 1 digit of precision for $s_{12} = 4$ after 192 hours. By avoiding contour deformation, the time to obtain this digit is therefore accelerated by a factor of $\frac{689858}{23} \approx 29994\times$. The cost to compute additional digits using contour deformation is prohibitive, however, for the resolved integral, we can obtain 6 digits of precision in about 100 seconds. For BNP6 we see a similar behaviour as for the other massless examples. For example, to obtain about 3 digits of precision with $s_{12} = 4000$ we see a factor of $\frac{3404}{4.8} \approx 709\times$ speed-up from avoiding contour deformation. To obtain about 6 digits of precision, but with $s_{12} = 4$ instead, avoiding contour deformation yields a factor of $\frac{92}{6.4} \approx 14\times$ speed-up.

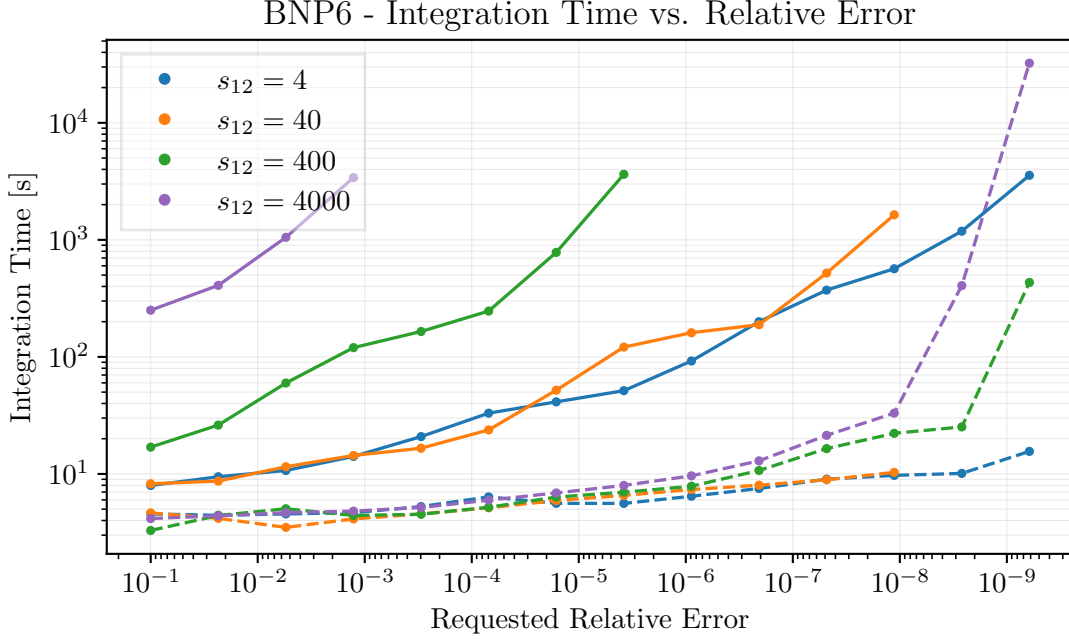


Figure 4.20.: Timings with (solid lines) and without (dashed lines) contour deformation for the 2-loop non-planar box with 6 propagators, expanded up to the finite order. Evaluated for different values of s_{12} with $s_{23} = -1$ fixed.

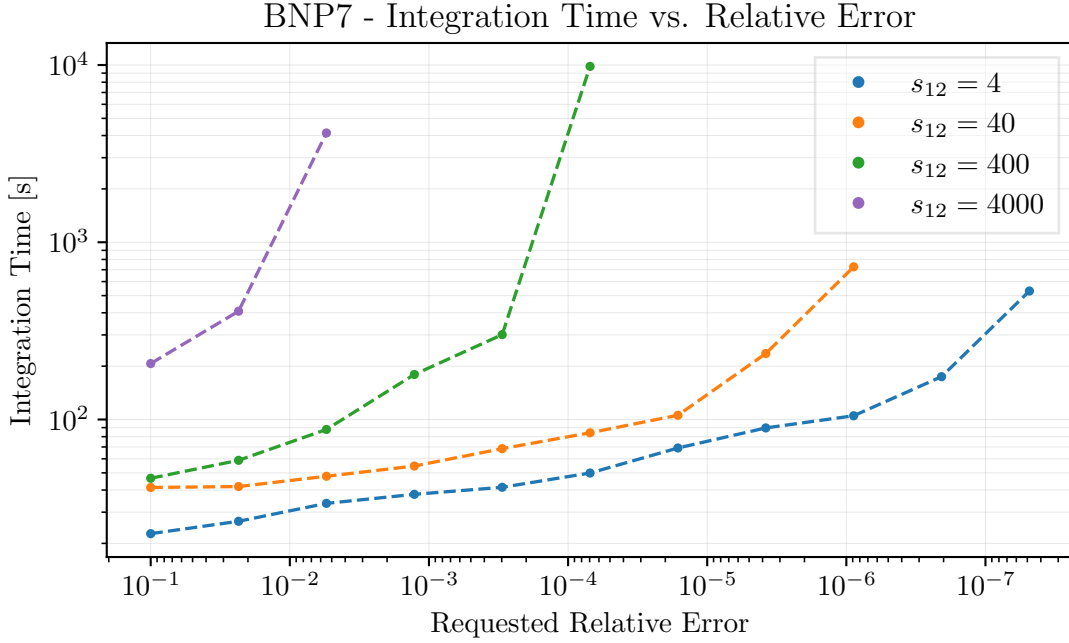


Figure 4.21.: Timings with (solid lines) and without (dashed lines) contour deformation for the 2-loop non-planar box with 7 propagators, expanded up to the finite order. Evaluated for different values of s_{12} with $s_{23} = -1$ fixed. In this example no digits could be obtained with contour deformation within a time limit of 10 hours.

4.4.4. 3-loop non-planar box

Figure 4.22 shows the integration times for the 3-loop non-planar box, only without contour deformation, for increasing values of the centre-of-mass energy s_{12} with $s_{13} = -1$ fixed. Additionally, it includes the point $s_{12} = 1, s_{13} = -0.2$, both with and without contour deformation, since this is the only test point we evaluated where a valid contour could be found with PYSECDEC. For this point, to obtain about 8 digits of precision, we observe a factor of $\frac{670}{62.7} \approx 11 \times$ speed-up from avoiding contour deformation. For all other points, integrating with contour deformation fails to provide any result while the resolved integrals achieve high precision for the full range in s_{12} . This is different from BNP7, since there valid contours are found but the integration instead converges too slowly to yield any digits. This means that no amount of computing resources would yield results for the 3-loop box when integrating with contour deformation (as implemented in PYSECDEC).

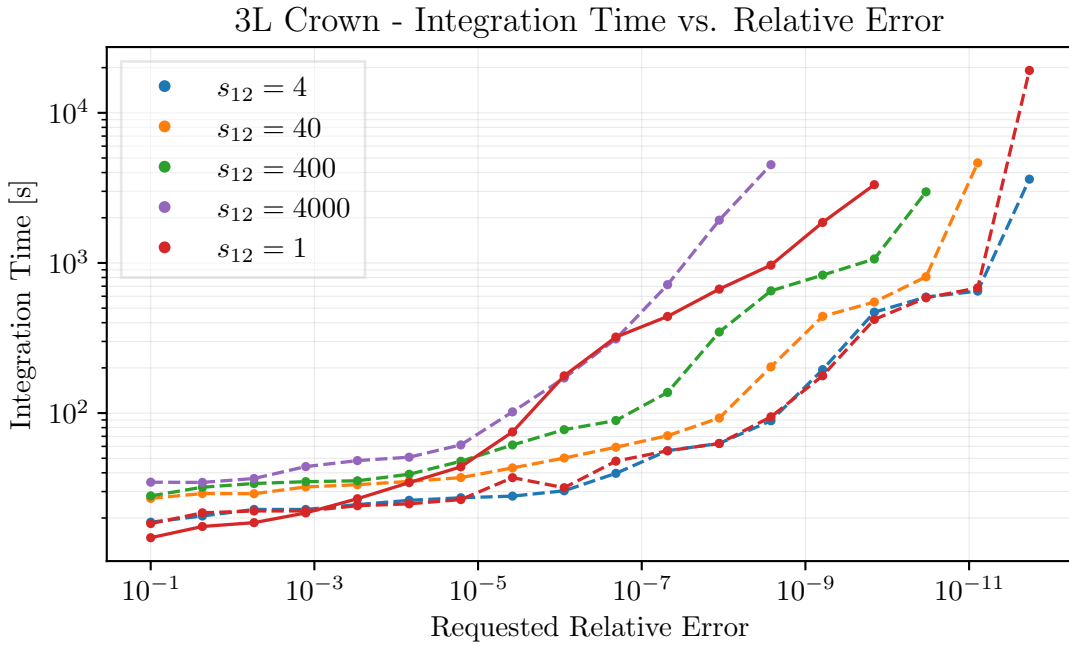


Figure 4.22.: Timings with (solid lines) and without (dashed lines) contour deformation for the 3-loop non-planar box, with only the leading ϵ^{-4} pole included. Evaluated for different values of s_{12} with $s_{13} = -1$ fixed, except for the point $s_{12} = 1$, where $s_{13} = -0.2$. For the benchmarks where $s_{12} \geq 4$ no digits could be obtained with contour deformation.

4.4.5. 1-loop triangle with an off-shell leg

Figure 4.23 shows the integration times for the 1-loop triangle, with and without contour deformation, for increasing values of the centre-of-mass energy (parameterised by β in this case), with the internal mass fixed as $m^2 = 1$. For most phase-space points, the triangle can be evaluated to high precision in short time, both with and without contour deformation. The minimal lattice size of the QMC integrator is almost enough to reach about 10 digits of precision, and the example is essentially too simple to see any gain from removing contour deformation. In fact, the integrand evaluations are not the bottleneck of this integral, which motivates us to look strictly at the integration time for the benchmarks, and ignore the overhead of loading prefactors. Despite these measures, we still see that the resolved version of the integral evaluates more slowly than when using contour deformation. The reason is that, as a consequence of our chosen resolution procedure, the resolved integrands have singularities at both the upper and the lower integration boundary. To handle this, PYSECDEC splits the integrals and remaps all singularities to the lower boundary, which results in more sectors and thereby increased evaluation time. For the positive contributions, $J_{\text{tri}}^{+,1}$, $J_{\text{tri}}^{+,2}$ and $J_{\text{tri}}^{+,3}$, the singularities are only present at one of the boundaries, and the splitting could either be turned off, or avoided by applying simple transformations of the type $x_i \rightarrow 1 - x_i$ before passing them to PYSECDEC. This reduced the number of sectors significantly, and brought the integration time of the resolved integral close to the time of integrating with contour deformation. For the negative contribution, J_{tri}^- , singularities are present at both boundaries, and splitting the integral is therefore necessary. The small remaining time difference can mostly be attributed to the extra sector this split generates. Usually, dealing with extra sectors due to the resolution procedure is not a major issue, since the resolved integrands typically scale much better. The problem with the triangle, is that it is too easy to integrate in either case, and the scaling with respect to number of lattice points becomes irrelevant. In the other massive examples we also require integral splitting, but then the contour deformed integrals are eventually unable to reach higher precision, and we are able to access more digits with the resolved integrands.

The situation is different when looking at the small-mass regime. The three-mass triangle integral is finite, however, in the limit $m^2 \rightarrow 0$ the integral develops poles in ϵ up to $1/\epsilon^2$. Therefore, when the integral is evaluated with small mass values, we numerically approach an end-point singularity which pinches the contour close to the boundaries of integration. And indeed, we observe in Figure 4.24 that integrating with contour deformation gets progressively worse for smaller values of the internal mass while the resolved integral is stable across all values of m^2 . For example, already for $m^2 = 10^{-6}$ we observe a factor of $\frac{368}{0.45} \approx 818 \times$ speed-up when avoiding contour deformation. For smaller values of m^2 the difference is even larger already for lower precision levels. For $m^2 = 10^{-12}$ we almost get no convergence with contour deformation while the resolved integral remains mostly stable. This demonstrates that the resolution procedure can be useful for simple examples too, albeit more so in extreme kinematic configurations.

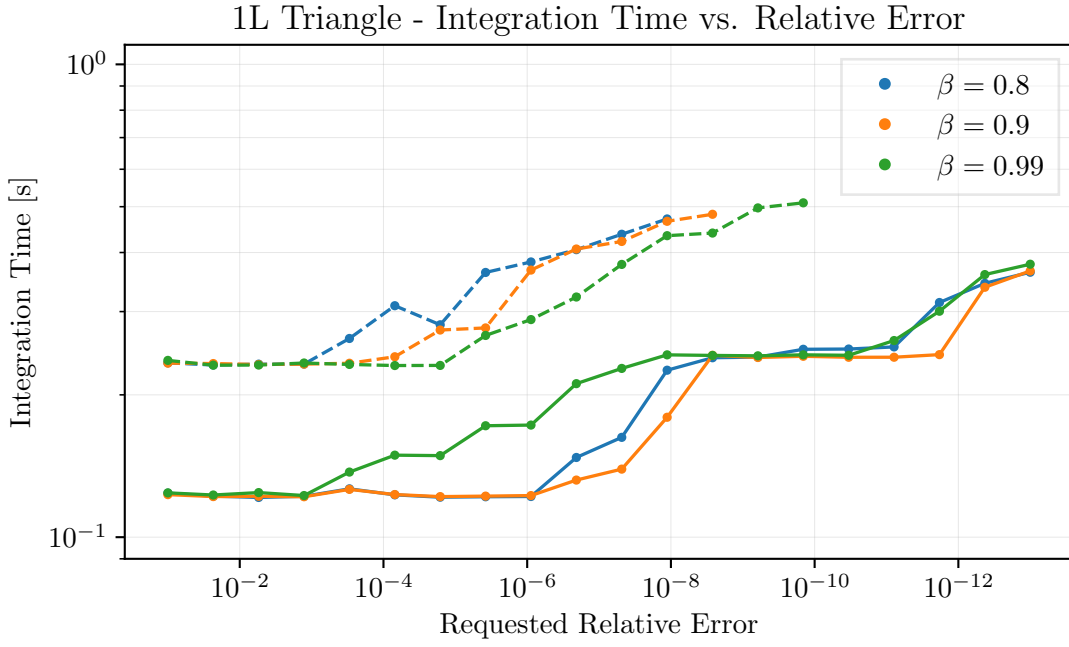


Figure 4.23.: Timings with (solid lines) and without (dashed lines) contour deformation for the all-massive 1-loop triangle, expanded up to order ϵ^4 . Evaluated for different values of β with $m = 1$ fixed.

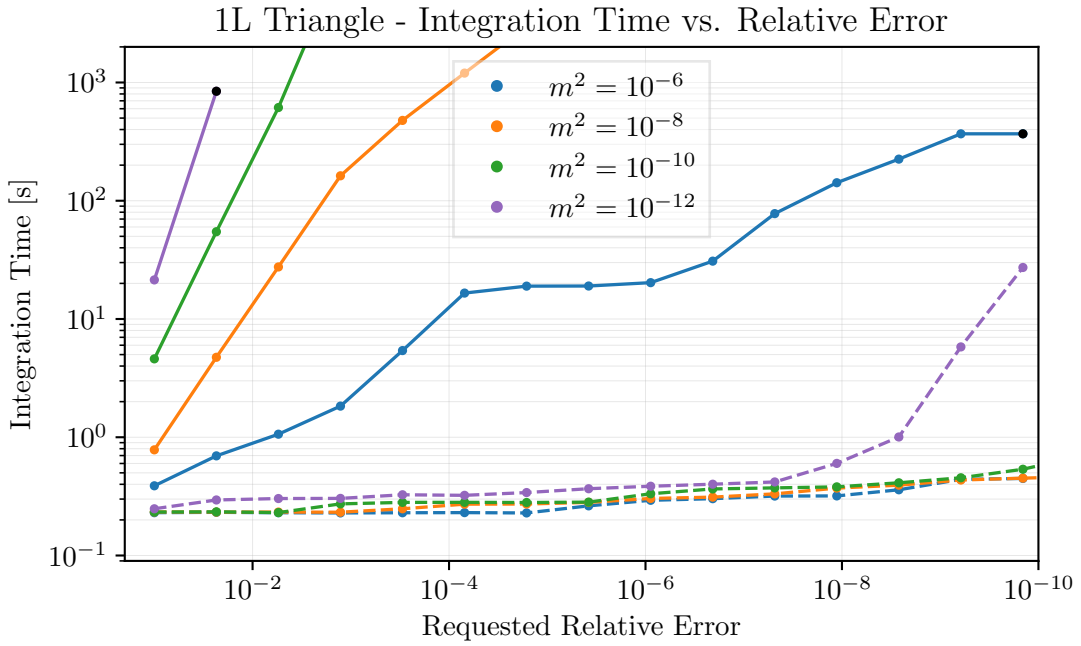


Figure 4.24.: Timings with (solid lines) and without (dashed lines) contour deformation for the all-massive 1-loop triangle, expanded up to order ϵ^4 . Evaluated for different values of m^2 with $s_{12} = 1$ fixed. The black dots indicate that no higher accuracy could be reached within 5 hours.

4.4.6. 2-loop elliptic sunrise

Figure 4.25 shows the integration times for the 2-loop massive sunrise, with and without contour deformation, for increasing values of the centre-of-mass energy β with $m = 2$ fixed. Figure 4.26 shows the same but for smaller values of the internal mass m , keeping $s = 1$ fixed instead. This integral is similar to the massive triangle in that the integration time for a small lattice size is not much faster without contour deformation. The difference here is that for a fixed lattice size, using contour deformation does not yield the same precision as the resolved integrals do. This hints at the variance of the resolved integrands being smaller than the variance of the contour deformed integrands. This is more clearly demonstrated in Table 4.1 where the actual relative errors at different QMC lattice sizes are displayed. The relative errors are estimated by:

$$\varepsilon_{\text{rel}} = \sqrt{\frac{(\text{Re}(\varepsilon_{\text{abs}})^2 + \text{Im}(\varepsilon_{\text{abs}})^2)}{(\text{Re}(I)^2 + \text{Im}(I)^2)}}, \quad (4.149)$$

where ε_{abs} is the absolute error estimated by the QMC integrator from random shifts, and I is the QMC estimate for the integral. The appearance of the timing figures can be understood by looking at the results for the higher ϵ -orders. The results from the contour deformed integrals reach a minimum error that is several digits larger than the results from the resolved integrals, and increasing the lattice size fails to improve the situation. This means that besides improving integration time, the new procedure additionally allows us to obtain more digits than could ever be obtained when integrating with contour deformation.

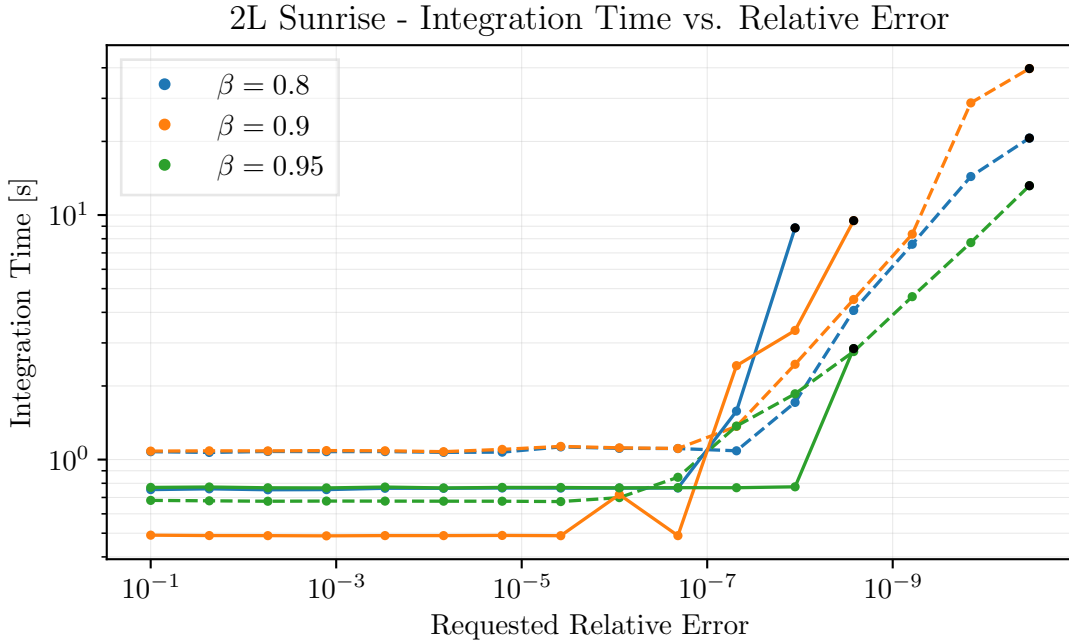


Figure 4.25.: Timings with (solid lines) and without (dashed lines) contour deformation for the all massive 2L elliptic sunrise, expanded up to order ϵ^4 . Evaluated for different values of β with $m = 2$ fixed. The black dots indicate points at which the evaluation time exceeded 10 hours.

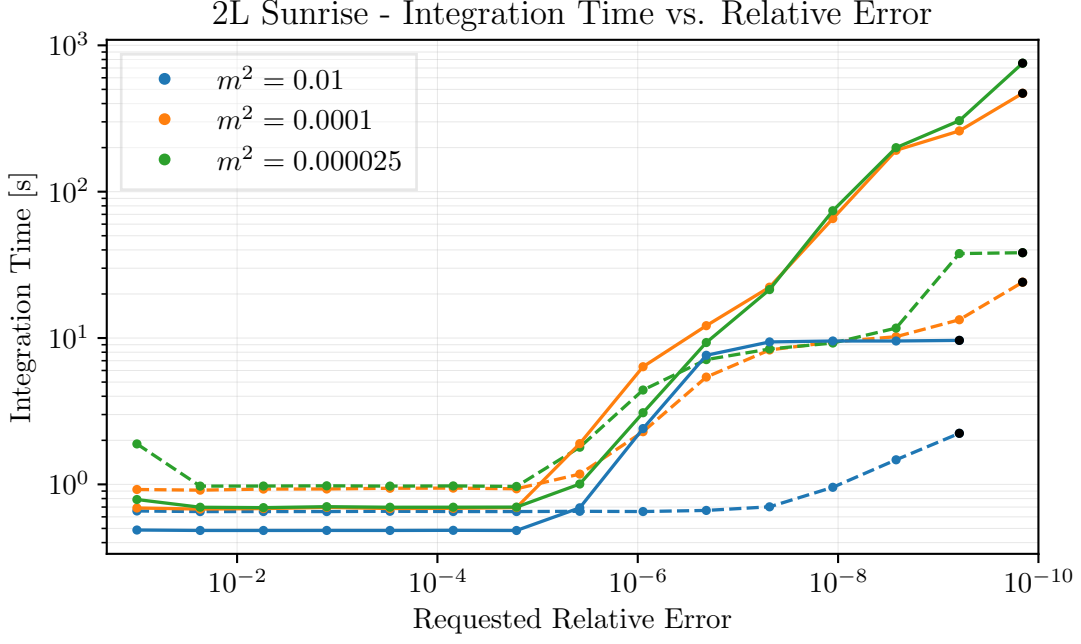


Figure 4.26.: Timings with (solid lines) and without (dashed lines) contour deformation for the all massive 2L elliptic sunrise, expanded up to order ϵ^4 . Evaluated for different values of m^2 with $s_{12} = 1$ fixed. The black dots indicate points after which the evaluation time diverged, and the integration was terminated after > 10 hours.

ϵ -order \ points		10^3	10^4	10^5	10^6
ϵ^0	CD	$3.41 \cdot 10^{-10}$	$4.84 \cdot 10^{-14}$	$5.86 \cdot 10^{-15}$	$8.39 \cdot 10^{-16}$
	No CD	$3.13 \cdot 10^{-10}$	$2.47 \cdot 10^{-14}$	$1.72 \cdot 10^{-16}$	$3.35 \cdot 10^{-16}$
ϵ^1	CD	$1.11 \cdot 10^{-9}$	$2.87 \cdot 10^{-13}$	$2.10 \cdot 10^{-13}$	$1.08 \cdot 10^{-13}$
	No CD	$1.13 \cdot 10^{-9}$	$8.91 \cdot 10^{-14}$	$6.53 \cdot 10^{-14}$	$9.88 \cdot 10^{-14}$
ϵ^2	CD	$3.49 \cdot 10^{-9}$	$1.86 \cdot 10^{-12}$	$7.57 \cdot 10^{-13}$	$5.60 \cdot 10^{-12}$
	No CD	$3.24 \cdot 10^{-9}$	$2.87 \cdot 10^{-13}$	$4.44 \cdot 10^{-13}$	$7.99 \cdot 10^{-13}$
ϵ^3	CD	$9.22 \cdot 10^{-9}$	$9.02 \cdot 10^{-11}$	$7.19 \cdot 10^{-10}$	$2.47 \cdot 10^{-8}$
	No CD	$6.91 \cdot 10^{-9}$	$6.05 \cdot 10^{-12}$	$1.88 \cdot 10^{-12}$	$5.30 \cdot 10^{-12}$
ϵ^4	CD	$1.48 \cdot 10^{-8}$	$8.75 \cdot 10^{-10}$	$4.44 \cdot 10^{-7}$	$1.33 \cdot 10^{-8}$
	No CD	$7.90 \cdot 10^{-9}$	$1.77 \cdot 10^{-11}$	$5.31 \cdot 10^{-12}$	$1.47 \cdot 10^{-11}$

Table 4.1.: Relative errors for the massive 2-loop sunrise integral with and without contour deformation for different QMC lattice sizes, at orders ϵ^0 , ϵ^1 , ϵ^2 , ϵ^3 and ϵ^4 , with kinematics $\beta = 0.8$ and $m = 2$.

4.4.7. 3-loop hyperelliptic banana

Figure 4.27 shows the integration times for the 3-loop massive banana, with and without contour deformation, for increasing values of the centre-of-mass energy β with $m = 2$ fixed. Figure 4.28 shows the same but for smaller values of the internal mass m , keeping $s = 1$ fixed instead. For this integral we essentially observe the same results as for the massive sunrise, with the integration times for initial lattice sizes even being slightly faster for the resolved integrands in this case. Again, we obtain more digits than what is possible with contour deformation, as is demonstrated clearly in Table 4.2. The situation is the same in the small-mass regime, where we observe that the resolved integrals obtain many more digits than contour deformation manages before hitting the machine precision limit.

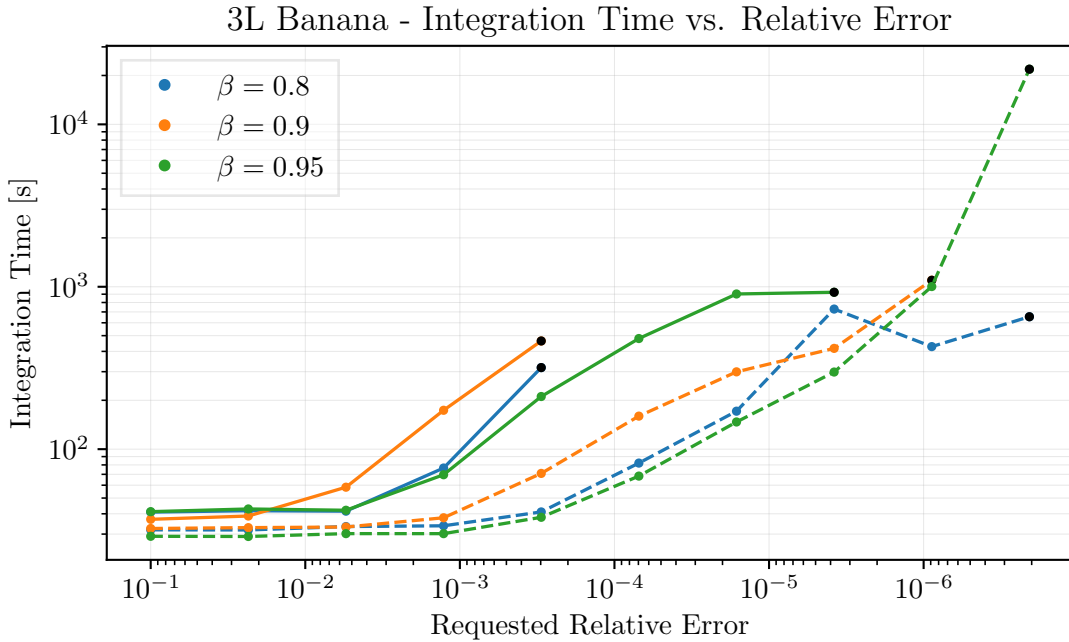


Figure 4.27.: Timings with (solid lines) and without (dashed lines) contour deformation for the all massive 3L banana, expanded up to order ϵ^4 . Evaluated for different values of β with $m = 2$ fixed. The black dots indicate points after which the evaluation time diverged, and the integration was terminated after > 10 hours.

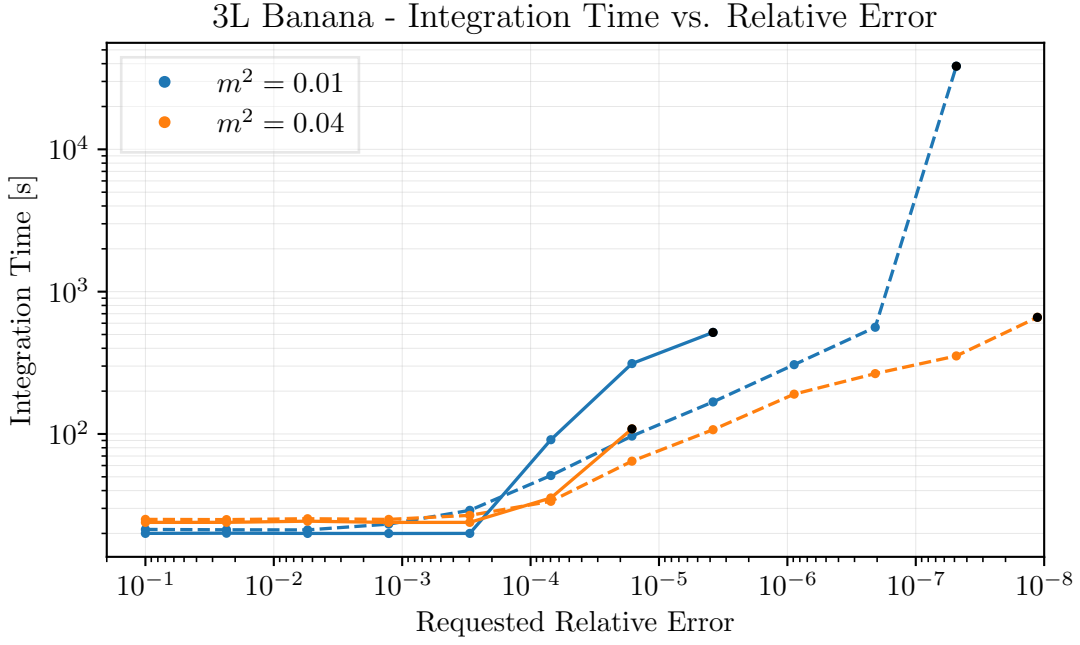


Figure 4.28.: Timings with (solid lines) and without (dashed lines) contour deformation for the all massive 3L banana, expanded up to order ϵ^4 . Evaluated for different values of m^2 with $s_{12} = 1$ fixed. The black dots indicate points after which the evaluation time diverged, and the integration was terminated after > 10 hours.

ϵ -order \ points		10^3	10^4	10^5	10^6
ϵ^0	CD	$2.37 \cdot 10^{-4}$	$1.43 \cdot 10^{-8}$	$7.80 \cdot 10^{-11}$	$1.96 \cdot 10^{-11}$
	No CD	$3.92 \cdot 10^{-5}$	$2.88 \cdot 10^{-8}$	$5.96 \cdot 10^{-10}$	$7.79 \cdot 10^{-10}$
ϵ^1	CD	$6.19 \cdot 10^{-4}$	$3.81 \cdot 10^{-8}$	$5.15 \cdot 10^{-9}$	$1.34 \cdot 10^{-8}$
	No CD	$1.01 \cdot 10^{-4}$	$8.90 \cdot 10^{-8}$	$1.44 \cdot 10^{-9}$	$3.15 \cdot 10^{-9}$
ϵ^2	CD	$9.26 \cdot 10^{-4}$	$1.41 \cdot 10^{-6}$	$6.06 \cdot 10^{-6}$	$1.06 \cdot 10^{-5}$
	No CD	$1.86 \cdot 10^{-4}$	$1.98 \cdot 10^{-7}$	$1.67 \cdot 10^{-8}$	$1.18 \cdot 10^{-8}$
ϵ^3	CD	$1.51 \cdot 10^{-3}$	$6.89 \cdot 10^{-6}$	$4.89 \cdot 10^{-5}$	$7.21 \cdot 10^{-5}$
	No CD	$3.69 \cdot 10^{-4}$	$4.70 \cdot 10^{-7}$	$5.64 \cdot 10^{-8}$	$4.98 \cdot 10^{-7}$
ϵ^4	CD	$5.05 \cdot 10^{-3}$	$1.46 \cdot 10^{-3}$	$5.21 \cdot 10^{-4}$	$2.36 \cdot 10^{-2}$
	No CD	$4.74 \cdot 10^{-4}$	$1.17 \cdot 10^{-6}$	$6.16 \cdot 10^{-7}$	$6.41 \cdot 10^{-7}$

Table 4.2.: Relative errors for the hyperelliptic banana integral with and without contour deformation for different QMC lattice sizes, at orders ϵ^0 , ϵ^1 , ϵ^2 , ϵ^3 and ϵ^4 , with kinematics $\beta = 0.8$ and $m = 2$.

4.4.8. Cancellations

Since the resolved integrals are split into a sum of several contributions, there could, in principle, be a drop in precision on the full integrals due to cancellations. The benchmarks in the previous section already take this into account by considering the relative error on the full integral²⁸. We therefore already implicitly know that if there are any cancellations, they are not severe for the kinematic configurations considered so far. In this section, we demonstrate that severe cancellations do not occur for non-zero terms in the ϵ expansion of the integrals as we scan over the phase space. However, our resolution procedure can (in fact, must in some cases) generate spurious poles in ϵ , which then cancel upon summation. These spurious singularities nevertheless do not degrade the performance of our resolved integrals as significantly as using contour deformation.

In Figure 4.29, we show the positive and negative contributions to the real parts of the ϵ^{-3} and ϵ^0 orders of BNP6 (massless 2-loop non-planar box), for values of $s_{12} \in (4, 100)$ with $s_{23} = -1$ fixed. For the ϵ^0 coefficient, the cancellation between the positive and negative pieces is relatively small, remaining $\mathcal{O}(1)$ for most points. In the small s limit, where we approach the limit of the validity of our resolution, the integral is dominated by J_{BNP6}^+ , with no significant cancellation occurring between the different terms. Instead, in the high-energy limit for $s \sim 100$, we do observe cancellations of 1 digit of precision. The final result for the BNP6 integral has poles up to and including ϵ^{-2} . However, after resolution, the integral develops a spurious ϵ^{-3} pole which vanishes due to cancellation between the positive and negative contributions²⁹. The same type of cancellation happens for the finite massive triangle, where a spurious ϵ^{-1} pole appears and is cancelled due to $J_{\text{tri}}^{+,2}$ and J_{tri}^- , which can be seen in Figure 4.30. One can understand why this pole must appear by looking at the expansion of the prefactor to the negative contribution, which for the 1-loop triangle is

$$\lim_{\delta \rightarrow 0^+} (-1 - i\delta)^{-1-\epsilon} = -1 - i\pi\epsilon + \mathcal{O}(\epsilon^2). \quad (4.150)$$

We know that this integral is finite and complex at ϵ^0 . This means that J_{tri}^- must necessarily have a ϵ^{-1} contribution such that a complex part can be generated at leading order (the positive contributions are purely real). It follows that some of the positive contributions must also have a ϵ^{-1} contribution to cancel the pole and leave the result finite. Coefficients of spurious poles can not be numerically integrated to high relative precision (though good absolute precision can be obtained). For the benchmarks in this chapter, we skip the integration of the coefficients of poles that we know are spurious. In practice, the order of the leading pole might not be known, and in such cases, the integration must be terminated based on an absolute error instead. In future work, it would be interesting to explore to what extent the cancellation of spurious singularities between resolved regions can be arranged locally in parameter space. This research direction is simplified in our approach by the fact that we know these cancellations occur precisely on an integration boundary, and we retain knowledge of which boundary in each resolved integral contributes to the spurious singularity. Achieving a local cancellation would eliminate the appearance of spurious poles, allowing the finiteness of

²⁸This is automated within the `sum_package` module of PYSECDEC.

²⁹In fact, for this integral, additional spurious ϵ^{-3} poles appear also during the sector decomposition of the original integral when using contour deformation.

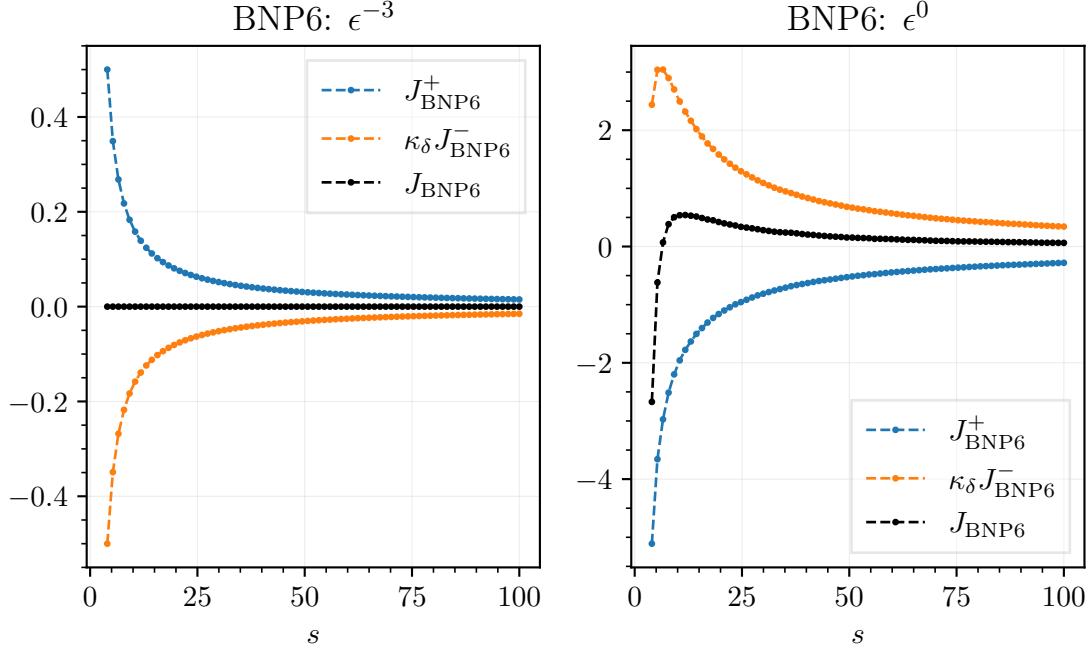


Figure 4.29.: Magnitude of the real part of the positive and negative contributions compared to the total integral for BNP6 at orders ϵ^{-3} and ϵ^0 . $\kappa_\delta = \lim_{\delta \rightarrow 0^+} (-1 - i\delta)^{-2-2\epsilon}$. The ϵ^{-3} pole is spurious and is a consequence of the cancellation between J_{BNP6}^+ and $\kappa_\delta J_{\text{BNP6}}^-$. $s \in (4, 100)$ and $s_{23} = -1$ fixed.

integrals to be manifestly maintained and removing any associated performance impact. We remark that this situation is potentially related to the more general problem of the appearance of spurious poles in the context of sector decomposition.

Figure 4.30 also shows the interplay between the contributions at orders ϵ^0 , ϵ^3 and ϵ^4 for the 1-loop triangle integral. We see explicitly that there are no severe cancellations in this case either. Finally, Figure 4.31 shows the magnitude of the one negative and three positive contributions to the 2-loop massive sunrise. In this case we observe that at each order the value of the full integral diverges as $\beta \rightarrow 1$. However, there are still no significant cancellations between the resolved pieces. Additionally, we include the result from using contour deformation to demonstrate that it agrees at each point with the sum of the resolved contributions, also in the high-energy regime where contributions become very large.

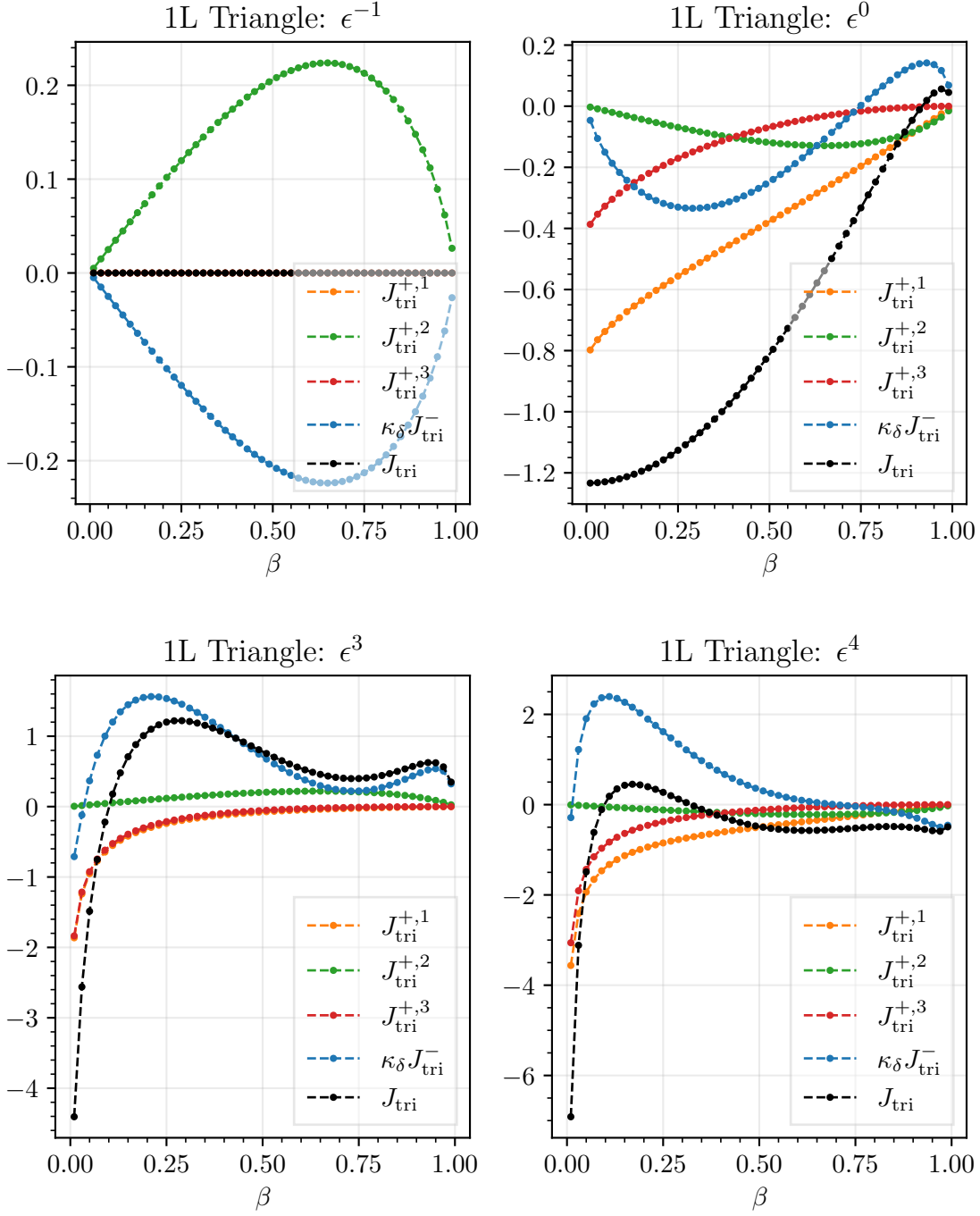


Figure 4.30.: Magnitude of the real part of the positive and negative contributions compared to the total integral for the all massive 1-loop triangle at orders ϵ^{-1} , ϵ^0 , ϵ^3 and ϵ^4 . $\kappa_{\delta} = \lim_{\delta \rightarrow 0+} (-1 - i\delta)^{-1-\epsilon}$. The 1, 2, 3 indices corresponds to the different positive regions shown in Figure 4.10. $\beta \in (0.01, 0.99)$ and $m = 1$ fixed.

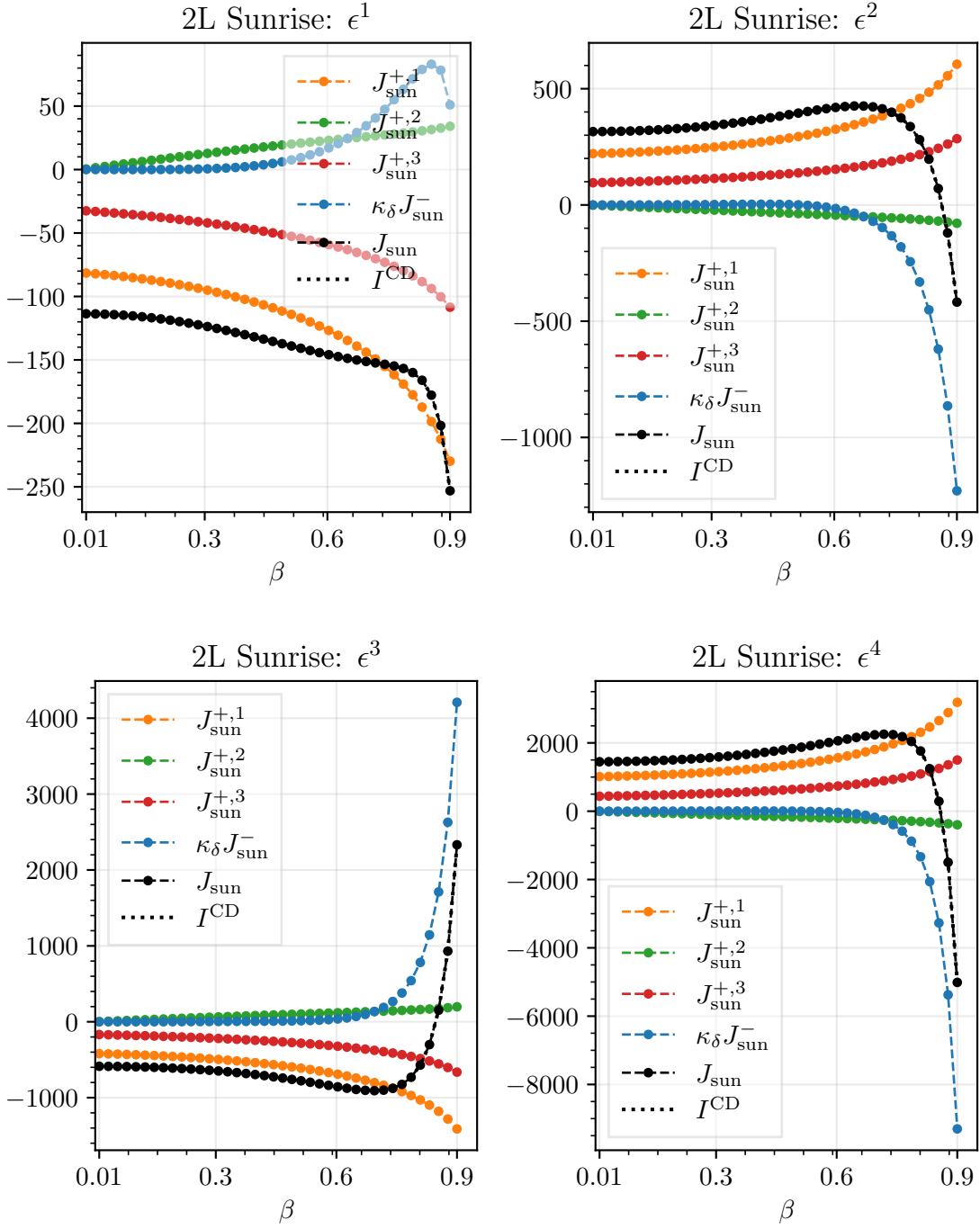


Figure 4.31.: Magnitude of the real part of the positive and negative contributions compared to the total integral for the elliptic sunrise at orders 1, 2, 3, 4 in the ϵ expansion. $\kappa_\delta = \lim_{\delta \rightarrow 0^+} (-1 - i\delta)^{-1-2\epsilon}$. The 1, 2, 3 indices corresponds to the different positive regions shown in 4.13. $\beta \in (0.01, 0.9)$ and $m = 2$.

4.5. Conclusion and outlook

The method presented in this chapter, as demonstrated in Sec. 4.4, is immediately relevant for numerical approaches relying on the use of parameter space. So far, the focus has been on the impact in the context of sector decomposition-based approaches; however, this concept is also immediately relevant for avoiding contour deformation in tropical geometry approaches to computing Feynman integrals [205, 206]. Other numerical approaches, such as those using integrable neural networks to approximate the integrand [207], which are known to work for Euclidean integrals but are not as straightforward to apply to integrals in the Minkowski regime, are also likely to benefit from this method.

Other, quite conceptually different, numerical or semi-numerical methods for computing Feynman integrals are also plagued or completely obstructed by the presence of physical thresholds. For example, in Ref. [208], an approach to approximating/computing Feynman integrals using positivity constraints was presented. The method was demonstrated to perform well for integrals in the Euclidean region, relying on the fact that the integrand is guaranteed to have the same sign in this region. Our work suggests that any Feynman integral (even those without a Euclidean region) can be cast in this form; therefore, it would be interesting to investigate the performance of this technique on our integrands. In this context, we emphasise that it would be essential (or at least extremely helpful) to have a robust implementation of integration-by-parts identities directly in parameter space [109, 209–212]. In Ref. [213], a method was introduced to compute Feynman integrals both above and below thresholds based on Taylor expansion and subsequent integration in parameter space. The method applies to a wide variety of integrals and avoids the need to numerically integrate the parameter space representation; however, as demonstrated by the authors of that work, the performance in some cases is significantly reduced by the presence of thresholds. We speculate that this large performance impact is caused by the presence of $\mathcal{F} = 0$ type singularities, which our procedure would remove.

Finally, we observe that the representation we target, see Eq. (4.2), makes the analytic continuation of any integral manifest; it is controlled entirely by a factor of $(-1 - i\delta)$ raised to the power of the second Symanzik polynomial, or, equivalently, an overall phase for each Feynman integral. If we imagine writing an entire amplitude in terms of real resolved integrals, then the analytic continuation of any scattering process becomes trivial [214]. Furthermore, at the leading order in the ϵ expansion for finite integrals, the integrands appearing in our resolution are strictly non-negative. It would be interesting to explore if being able to obtain such a representation for a complete amplitude has any connection to concepts such as positive geometries, see e.g. Refs. [215, 216], and if any of the additional requirements important to this programme, such as complete monotonicity, could be satisfied. In Ref. [217], it was argued that taking $\mathcal{F}(\mathbf{x}; \mathbf{s}) = 0$ as a boundary of integration leads to maximal (and non-maximal) cuts/discontinuities, in general, it would be interesting to explore the interpretation of the integrals obtained from our approach. We also remark that, naively, a modification of the resolution procedure could be applicable in the context of the Method of Regions [31, 218]. Specifically, it has been shown that dissecting the domain of integration on the Landau singularities can reveal regions that are not visible using geometric methods based on Newton polytopes, which neglect the possibility of cancellations between monomials [81, 219]

whereas our method can naturally account for such cancellations by construction. Having a systematic method for mapping mixed-sign polynomials to same-sign polynomials for which the geometric approach to the Method of Regions applies provides a potentially useful step in fully systematising the Method of Regions in parameter space.

CHAPTER 5

Interpolating Amplitudes

The purpose of computing multi-loop virtual amplitudes is to increase the precision with which hard cross sections are calculated. If the IR subtraction terms are available, as well as the matching to parton showers, the multi-loop result can be used by general purpose MC event generators to simulate a full collider event. If the shower matching is not available, as is the case for many NNLO processes, the multi-loop amplitude can instead be used by partonic event generators, such as NNLOJET [220], MATRIX [221] and MCFM [222], to compute fixed-order cross sections. In either case, the availability of the multi-loop results are important as they serve as inputs to the MC program and determine features for the remainder of the simulation or calculation.

In the previous chapters both tools and techniques that facilitate multi-loop computations have been discussed in great detail. In Sec. 3.2 an overview of a standard workflow for computing multi-loop amplitudes was introduced. Depending on how the integrals are evaluated, the result is on either of the following two main forms³⁰

- Analytical: The amplitude is expressed as a linear combination of special functions.
- Numerical: A numerical value for each ϵ -order of the amplitude is known for some configuration of the kinematic invariants.

Neither of these results are straightforward to implement in a MC code. In the first case, a method for fast evaluation of the special functions must be achieved. This can be complicated if the function class is elliptic or beyond. In the second case, the amplitude must be re-evaluated for each required kinematic configuration. Due to importance sampling, it is usually not possible to predict which points will be requested,

³⁰A different approach is to compute the multi-loop amplitude in an approximation, such as in analytic expansions [168]. Combining approximations that are valid for different phase-space regions has recently been shown to be a useful approach [223, 224]. These methods provide a valuable complement to exact calculations in cases where the analytic form is not known or where numerical evaluations are too slow.

which prohibits precomputation. To the three-step process of computing a multi-loop amplitude, which was outlined in Sec. 3.2, it is therefore natural to include a fourth step; exporting the results.

A method of making multi-loop results available is the usage of interpolation frameworks. The idea is to precompute the amplitude at a small set of grid points and interpolate for the values in between. Such a procedure has been used successfully for several two-loop amplitudes already, such as those entering $t\bar{t}$ -production at NNLO [225, 226], HH production in gluon fusion at NLO [227, 228] and H +jet production at NLO [229, 230]. The two-loop amplitudes entering these $2 \rightarrow 2$ processes depend on just two kinematic variables once the masses have been fixed to their SM values. In contrast, the interpolation for $2 \rightarrow 3$ processes needs to be performed on a five-dimensional phase space, which makes achieving precision a highly non-trivial task. A strong motivation for this investigation is our recent work on the two-loop $t\bar{t}H$ amplitudes [16]. The extreme complexity of the master integrals necessitate numerical evaluations to obtain exact results. Ref. [16] is a proof-of-concept calculation for the quark-initiated part of the amplitude that includes diagrams containing quark loops (the N_f part). The full amplitude depends on many more master integrals, and evaluating it numerically is computationally very demanding. For this application, it is therefore crucial to have an as efficient approximation framework as possible, in order to reduce the required number of CPU/GPU hours.

This chapter presents an investigation of several state-of-the-art multidimensional interpolation methods applied to amplitudes [15]. In Sec. 5.1 we start by defining the interpolation problem. We introduce a framework to efficiently and accurately estimate how the approximation error propagates to observables. In Secs. 5.2 – 5.4 we describe classical interpolation methods based on polynomials, B-splines and sparse grids, and in Sec. 5.5 we describe regression-based machine learning techniques (MLP and L-GATr). At the end of these sections, we comment on the performance of the tested methods. In Sec. 5.6 we conclude by summarizing the discussion from the previous sections.

5.1. Setting the stage

The objective is to approximate a squared amplitude³¹

$$|\mathcal{M}|^2 \equiv \mathcal{A} = \sum_{k=0}^{\infty} \left(\frac{\alpha_s}{2\pi} \right)^k \mathcal{A}_k, \quad (5.1)$$

which is truncated at some order, say $k = 2$, such that $\mathcal{A} \approx \mathcal{A}_0 + \left(\frac{\alpha_s}{2\pi} \right) \mathcal{A}_1 + \left(\frac{\alpha_s}{2\pi} \right)^2 \mathcal{A}_2$. The assumption is that \mathcal{A}_2 is much more computationally demanding than either \mathcal{A}_0 or \mathcal{A}_1 , such that we can consider them readily available. The problem therefore reduces to approximating \mathcal{A}_2 . Besides the squared amplitude, the cross section differential depends on the parameters that fully characterize the phase space. These are two for $2 \rightarrow 2$ processes, five for $2 \rightarrow 3$, nine for $2 \rightarrow 4$, etc. There is freedom in choosing these parameters; common choices are energy or Mandelstam variables, $s_{ij} = (p_i + p_j)^2$, and angular variables. In what follows, we denote these parameters as \vec{x} , and choose them such that the physical region of the process is a unit hypercube: $\vec{x} \in (0; 1)^d$.

³¹We use α_s as the coupling since all test functions that appear later happen to be QCD processes.

The interpolation problem can now be formulated as follows; we aim to approximate a correction to a squared amplitude a by some function \tilde{a} , based on the knowledge of a at some *data points* $\vec{x}_1, \dots, \vec{x}_n$: $a_i \equiv a(\vec{x}_i)$. We are interested in algorithms to choose \vec{x}_i and to construct \tilde{a} such that it is “close enough” to a , while requiring as few data points as possible, as to minimize the expensive evaluations of a .

5.1.1. Defining the error

There are different ways to precisely define what “close enough” means, and no single definition works equally well for all use cases. We will consider the error on the observable to be the quantity of interest, since this is what an event generator will need to know in the end. It is not possible to simultaneously optimize for all observables and phase-space regions, so a choice of what to prioritize must be made. For the examples in this chapter, we focus on total cross sections and take $d\sigma/d\vec{x}$ to be the quantity of interest. We define the approximation error by viewing it as a probability density and using the distance between $d\sigma/d\vec{x}$ based on \tilde{a} and a , measured via the L^1 norm:³²

$$\varepsilon = \frac{\|\tilde{f} - f\|_1}{\|f\|_1}, \quad \text{where} \quad f(\vec{x}) \equiv a(\vec{x}) \underbrace{\frac{1}{2\hat{s}} \left| \frac{d(\Phi, \rho)}{d\vec{x}} \right|}_{\equiv \text{weight } w(\vec{x})}, \quad (5.2)$$

where $d\Phi$ is the element of the Lorentz-invariant phase space and $d\rho \equiv d\rho_{a,b}(\hat{s}, s)$ describes the probability of finding the incoming particles a and b with collision energy $\sqrt{\hat{s}}$. For example, in proton-proton collisions this denotes the contribution of the parton distribution functions. The choice of the L^1 norm ensures that this quantity is independent of the choice of variables \vec{x} (which would not be the case for e.g. L^2 norm). In statistics, ε is known as the *total variation distance* (up to an overall normalization). This distance weighs different phase-space regions proportional to their contribution to the total cross section (via the factor w), and guarantees that for any phase-space subregion R , using $\tilde{a}(\vec{x})$ instead of $a(\vec{x})$ will result in the error of $\int_{\vec{x} \in R} d\sigma$ being no more than $\varepsilon \left(\frac{\alpha_s}{2\pi}\right)^k \|f\|_1$.

Note that the precision of the total cross section comes at the expense of precision in tails of differential distributions: for parts of the phase space that do not contribute much to the total cross section, such as the very-high-energy region, only low precision is guaranteed by a bound on ε . An alternative definition of the error, $\varepsilon = \max |\tilde{a}/a - 1|$, would ensure equal relative precision for all bins of any differential distribution, but satisfying it would come at the expense of the interpolation spending most effort on phase-space regions where only few (or none at all) experimental data points are expected at the LHC. This is the trade-off involved in error target choices; to apply the methods we study, each application would need to choose its own appropriate error measure.

5.1.2. The test functions

In the remainder of this chapter we assess the performance of the most promising interpolation algorithms studied in numerical analysis when applied to the following five test functions.

³²Here and throughout, we define the L^p -norm $\|f\|_p$ as $(\int |f(\vec{x})|^p d\vec{x})^{1/p}$.

- f_1 : LO amplitude for $q\bar{q} \rightarrow t\bar{t}H$.
- f_2 : NLO amplitude contributing to $q\bar{q} \rightarrow t\bar{t}H$.
- f_3 : LO amplitude for $gg \rightarrow t\bar{t}H$.
- f_4 : NLO amplitude contributing to $gg \rightarrow t\bar{t}H$.
- f_5 : LO (one-loop) amplitude for $gg \rightarrow Hg$.

In each case, f_i is actually the cross section differential corresponding to the amplitudes listed above. The $t\bar{t}H$ test functions ($f_1 - f_4$) are taken as 5-dimensional functions over the phase space as described in Ref. [16]. For example, for f_1 we have

$$a_1 = \langle \mathcal{M}_0^{q\bar{q}t\bar{t}H} | \mathcal{M}_0^{q\bar{q}t\bar{t}H} \rangle, \quad f_1 = a_1 \times \left| \frac{d\Phi_{t\bar{t}H}}{d(\text{frac}_{s_{t\bar{t}}}, \theta_H, \theta_t, \varphi_t)} \right| \times \frac{1}{2\hat{s}} \frac{d\rho_{q\bar{q}}}{d\beta^2} \times J_{t\bar{t}H}, \quad (5.3)$$

with the phase space parameters set as

$$\beta^2 = \frac{10}{100} + \frac{86}{100}x_1, \quad \text{frac}_{s_{t\bar{t}}} = x_2, \quad \theta_H = \pi x_3, \quad \theta_t = \pi x_4, \quad \varphi_t = 2\pi x_5, \quad (5.4)$$

$$J_{t\bar{t}H} = \left| \frac{d(\beta^2, \text{frac}_{s_{t\bar{t}}}, \theta_H, \theta_t, \varphi_t)}{d\vec{x}} \right| = \frac{86}{50}\pi^3. \quad (5.5)$$

To specify $\rho_{q\bar{q}}$ we use the following generic form

$$\frac{1}{2\hat{s}} \frac{d\rho_{q\bar{q}}}{d\beta^2} \propto \frac{(1 - c_1\beta^2)^2}{(1 - c_2\beta^2)(1 - c_3\beta^2)}, \quad (5.6)$$

which approximates ρ computed using the parton distribution functions from Ref. [231]. For the parameters $c = \{c_1, c_2, c_3\}$ we use respectively for

$$\begin{aligned} f_1 \text{ and } f_2 : c &= \{1.0132, 0.9943, 0.3506\}, \\ f_3 \text{ and } f_4 : c &= \{1.0134, 0.7344, 0.0987\}, \\ f_5 : c &= \{1.0012, 0.9802, 0.3357\}. \end{aligned}$$

The overall proportionality constant is omitted here because it cancels in the error definition of Eq. (5.2). The remaining test functions are defined analogously to f_1 . These definitions, as well as details on the phase-space parameters, amplitude symmetries and illustrations of phase-space slices, can be found in App. A. We do point out here that the one-loop $t\bar{t}H$ amplitudes have an integrable Coulomb-type singularity in the corner where $\text{frac}_{s_{t\bar{t}}} \rightarrow 0$. The corresponding test functions (f_2 and f_4) are defined with this singularity subtracted. Similarly, the phase space for f_5 is defined with p_T -cuts to avoid divergences in the forward and backward regions, see App. A. In all cases GOSAM [232] is used to evaluate the amplitudes, and we set $m_H^2/m_t^2 = 12/23$.

5.1.3. Estimating the error

To evaluate Eq. (5.2) we use Monte Carlo integration. It can be formulated based on different kinds of testing samples:

$$\text{uniform:} \quad \vec{x}_1, \dots, \vec{x}_m \sim 1, \quad \varepsilon = \frac{\sum_{i=1}^m |(\tilde{a}_i - a_i)w_i|}{\sum_{i=1}^m |a_i w_i|}, \quad (5.7)$$

$$\text{partially unweighted:} \quad \vec{x}_1, \dots, \vec{x}_m \sim w, \quad \varepsilon = \frac{\sum_{i=1}^m |\tilde{a}_i - a_i|}{\sum_{i=1}^m |a_i|}, \quad (5.8)$$

$$\text{leading-order unweighted:} \quad \vec{x}_1, \dots, \vec{x}_m \sim a_{\text{LO}} w, \quad \varepsilon = \frac{\sum_{i=1}^m |(\tilde{a}_i - a_i)/a_{\text{LO},i}|}{\sum_{i=1}^m |a_i/a_{\text{LO},i}|}. \quad (5.9)$$

The first of these is natural for the selected variables \vec{x} , the second is important because it is independent of the choice of \vec{x} (and can be approximated by e.g. the RAMBO sampling technique [233]), and the third is important because it is oriented at the approximate probability distribution of physical scattering events, encoded in the leading-order amplitude, a_{LO} . While each sampling method must yield the same result asymptotically, in practice we are interested in using as few testing evaluations as possible. In Fig. 5.1 we compare the error convergence when using the different sampling methods for test function f_2 . Here we see that the error becomes stable for all methods after $m \sim 1000$ samples. Note that a uniform sample can be generated both via Monte Carlo, i.e. randomly, and with a low-discrepancy sequence (such as the Sobol sequence). In Fig. 5.1 we present results for both options. We observe that, even though sampling from a low-discrepancy sequence gives a more uniform coverage of the parameter space, we see only marginal improvements in the error convergence.

5.1.4. Adaptivity and symmetries

Summarizing Sec. 5.1.3, we approximate $a(\vec{x})$, and encode the relative importance of different phase-space regions into $w(\vec{x})$. But can we incorporate this importance information to improve the approximation procedure? There are multiple options:

- (1). Instead of constructing an approximation for $a(\vec{x})$ directly, we can construct an approximation $\tilde{f}(\vec{x})$ for $f(\vec{x}) \equiv a(\vec{x}) w(\vec{x})$, and then set $\tilde{a}(\vec{x}) = \tilde{f}(\vec{x})/w(\vec{x})$.
- (2). For methods based on regression, we can choose the data points x_i such that they cluster more in regions where $w(\vec{x})$ is greater.
- (3). For adaptive methods, we can approximate $a(\vec{x})$, but use the quantity $a w$ in the adaptivity condition, so that regions where $a w$ (and not just a) is approximated the worst are refined first.
- (4). We can try to find a variable transformation from \vec{x} to \vec{y} that cancels w , and interpolate in \vec{y} instead of \vec{x} . In other words, choose τ with $\vec{x} = \tau(\vec{y})$ such that $w(\vec{x}) |d\tau(\vec{x})/d\vec{y}| = 1$, so that $\int a w dx = \int a dy$.

One popular approach to construct such transformations is to approximate τ as its rank-1 decomposition: $\tau(\vec{y}) = \prod_i \tau_i(y_i)$. This is the basis of the VEGAS integration algorithm [234]. Higher-rank approximations are, of course, also possible. Another approach is based on *normalizing flows* [235]: it involves fitting a Jacobian of a specially-constructed transformation to $1/w$. This machine learning approach has been extensively used to develop improvements in importance sampling [236–240].

These approaches can be applied on top of any interpolation method, and we shall not discuss them further.³³

Moreover, when the test functions are symmetric under discrete transformations, there are multiple ways to take advantage of this:

- (i). Make the interpolant obey the same symmetries by construction.

³³In our very limited testing, we did not see major gains from using these.

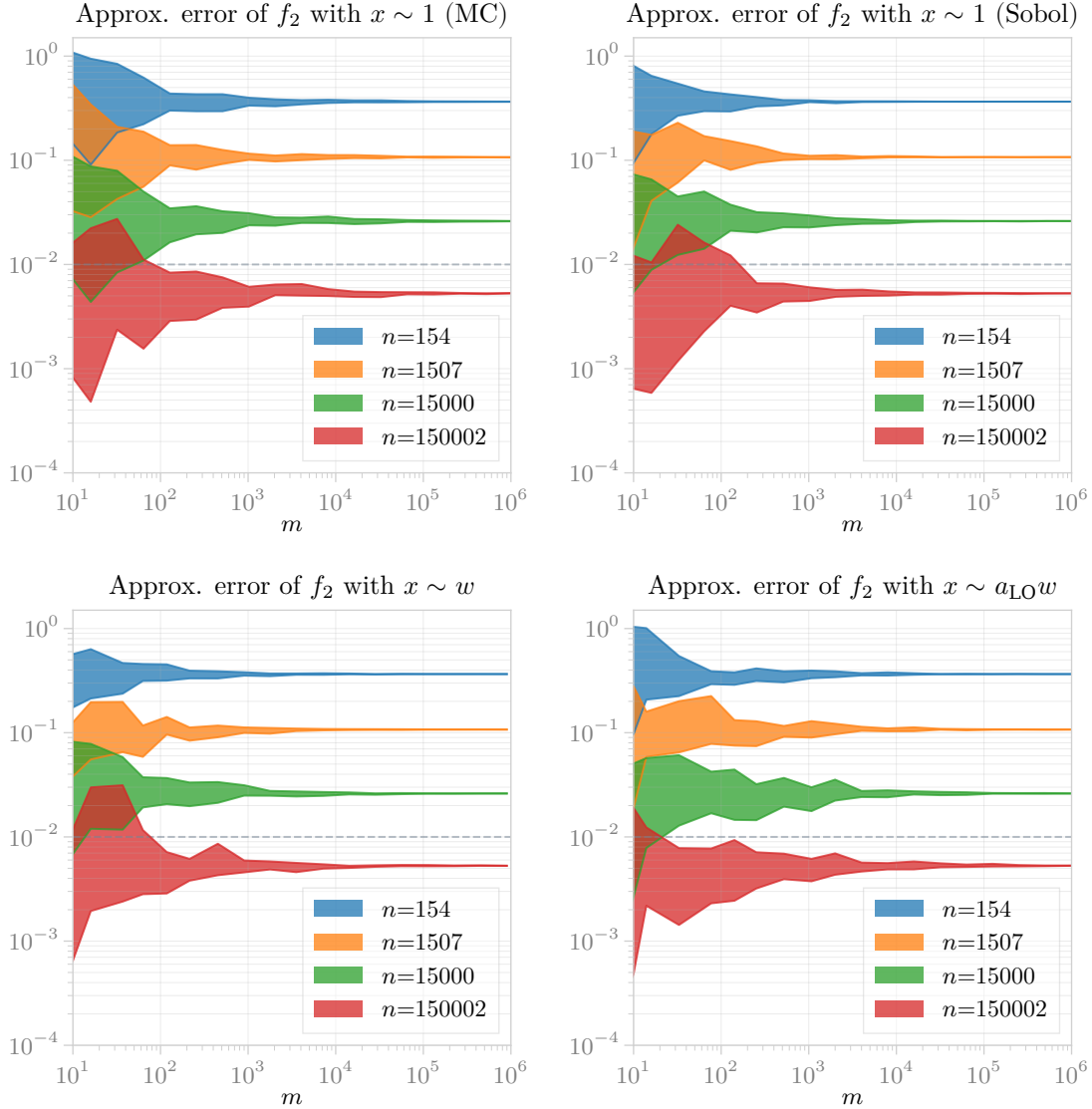


Figure 5.1.: Approximation error ε of f_2 , as defined in Eq. (5.2), evaluated via Eqs. (5.7) – (5.9), as a function of the number of testing points m . The uniform samples are taken both randomly (MC) and from a low-discrepancy sequence (Sobol). The approximations are constructed using sparse grid interpolation from Sec. 5.4, with different numbers of data points n . The error bands are created from 10 independent testing sets for each m .

- (ii). Duplicate symmetric data points, i.e. if $f(x) = f(1 - x)$, then for each x_i also add $1 - x_i$ to the data set (but still count this as a single evaluation of f).
- (iii). Reduce the interpolation domain, i.e. if $f(x) = f(1 - x)$, only construct the approximation for $x \in [0, 1/2]$, and use the symmetry to obtain the values in $x \in [1/2, 1]$.

5.2. Polynomial interpolation

In the following four sections, we describe four different approximation methods, starting with polynomial based interpolation [241, 242]. In the univariate case, \tilde{f} is constructed as a polynomial of degree $n - 1$ that passes through exactly n interpolation nodes x_i . It can be written in the *Lagrange form* as

$$\tilde{f}(x) = \sum_{i=1}^n f_i l_i(x), \quad l_i(x) \equiv \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}, \quad (5.10)$$

or slightly rewritten in the *barycentric form* [243] as

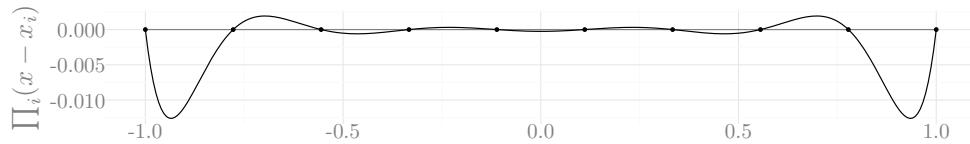
$$\tilde{f}(x) = \frac{\sum_{i=1}^n \frac{w_i}{x - x_i} f_i}{\sum_{i=1}^n \frac{w_i}{x - x_i}}, \quad w_i = \prod_{j \neq i} \frac{1}{x_i - x_j}. \quad (5.11)$$

The barycentric interpolation formula is general enough that any *rational* interpolation can be expressed by it through an appropriate choice of the *weights* w_i ; the weights given here, however, correspond to the purely polynomial interpolation of Eq. (5.10).³⁴ This is our form of choice for evaluation due to its numerical stability and simplicity.

The error of a polynomial approximation is given by

$$f(x) - \tilde{f}(x) = \frac{f^{(n)}(\xi)}{n!} \prod_{i=1}^n (x - x_i), \quad (5.12)$$

where ξ is some function of x . To minimize this error *a priori* without the precise knowledge of $f^{(n)}$, one can choose the nodes x_i such that they would minimize $\prod_i (x - x_i)$ over the domain of interest. Doing so is important because a naive choice of equidistant nodes leads to the *Runge phenomenon* [245]: the higher the degree of the polynomial, the worse the approximation error becomes close to the boundaries. E.g., for 10 nodes on the interval of $[-1; 1]$:



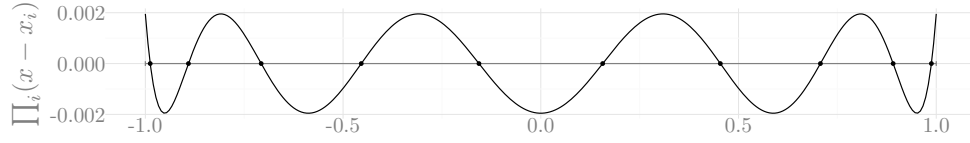
Chebyshev nodes of the first kind

The product $\prod_i (x - x_i)$ is minimized in the sense of the infinity norm by the *Chebyshev nodes of the first kind*, which are traditionally given for the domain $[-1; 1]$ as

$$x_i = \cos\left(\frac{2i - 1}{2n} \pi\right), \quad i = 1, \dots, n. \quad (5.13)$$

³⁴Rational interpolation methods, specifically of the non-linear kind, such as the AAA algorithm [244], have established themselves as the most efficient one-dimensional interpolation methods, but generalizations to many dimensions are not developed well enough for us to consider them.

These nodes achieve a uniform $\prod_i (x - x_i)$ over the interval:



Chebyshev polynomials

Corresponding to these nodes are the *Chebyshev polynomials of the first kind*:

$$T_k(x) = \cos(k \arccos(x)). \quad (5.14)$$

Specifically, Eq. (5.13) are the zeros of $T_n(x)$. These polynomials are orthogonal with respect to the weight $1/\sqrt{1-x^2}$:

$$\int_{-1}^1 \frac{T_n(x) T_m(x)}{\sqrt{1-x^2}} dx = \begin{cases} \pi & \text{if } n = m = 0, \\ \pi/2 & \text{if } n = m \neq 0, \\ 0 & \text{if } n \neq m. \end{cases} \quad (5.15)$$

Note that the nodes in Eq. (5.13) are nothing more than equidistant points in $\phi = \arccos(x)$, and the corresponding polynomials are simply an even Fourier series in ϕ . This is why a transformation from the function values $\{f_i\}$ to the coefficients of the decomposition into Chebyshev polynomials $\{c_i\}$,

$$\tilde{f}(x) = \sum_i c_i T_i(x), \quad (5.16)$$

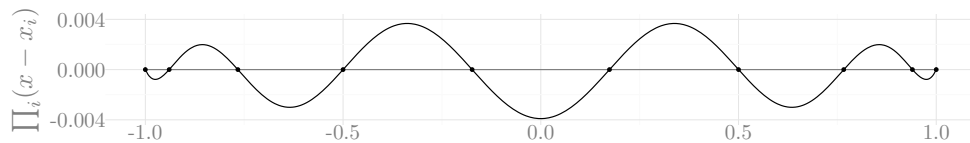
is just a Fourier transform (specifically, a discrete cosine transform). Still, for interpolation purposes, the barycentric form of Eq. (5.11) is preferable, since both Eq. (5.16) and the monomial form suffer from rounding errors that prevent their usage for $n \gtrsim 40$.

Chebyshev nodes of the second kind

A related set of points that avoids the Runge phenomenon is *Chebyshev nodes of the second kind* (a.k.a. *Chebyshev–Lobatto nodes*), traditionally given as

$$x_i = \cos\left(\frac{i-1}{n-1}\pi\right), \quad i = 1, \dots, n. \quad (5.17)$$

Unlike Eq. (5.13), these points are not located at the zeros of $T_n(x)$, but rather at the extrema and end points, with the advantage of them being nested: the set of n of these is exactly contained in the set of $2n-1$. This comes at the price of $\prod_i (x - x_i)$ not being uniform over the interval:



The property of being nested is important for adaptive interpolation constructions because larger grids can reuse the results of the smaller ones that they contain.

Unfortunately, the inclusion of the end points can make this construction impractical for scattering amplitudes. For example, f_2 can not be evaluated at exactly $x_2 = 0$ due to a loss of numerical precision, and evaluation at $x_1 = 1$ is possible but undesirable because evaluation time typically grows when approaching this boundary.

Gauss nodes

Closely related to Chebyshev nodes, and often considered superior, are *Gauss nodes* and *Gauss–Lobatto nodes*. These are defined respectively as the location of zeros and extrema of the Legendre polynomials. They have the advantage that a quadrature built on them (*Gauss quadrature*) is exact for polynomials up to degree $2n-1$, while the same for Chebyshev nodes (*Clenshaw–Curtis quadrature*) is only exact for polynomials up to degree $n-1$. In practice, however, the approximation error of both is very close [246, 247], and since Gauss nodes are much harder to compute compared to Eq. (5.13), we do not consider them further.

5.2.1. Approximation error scaling

It is known that polynomial interpolation at Chebyshev nodes is logarithmically close to the best polynomial interpolation of the same degree [241]. The approximation error itself depends on how smooth the function f is [242, 248, 249]. If f has $\nu-1$ continuous derivatives and the variation of $f^{(\nu)}$ is bounded, then

$$\|f - \tilde{f}\|_2 \leq \frac{4 \|f^{(\nu)}\|_1}{\pi \nu (n - \nu)^\nu}, \quad \text{for } n > \nu. \quad (5.18)$$

If f is analytic, and can be analytically continued to an ellipse in the complex plane with focal points at ± 1 and the sum of semimajor and semiminor axes ρ (a *Bernstein ellipse*), then

$$\|f - \tilde{f}\|_2 \leq \frac{4M\rho^{-n}}{\rho - 1}, \quad \text{where } M = \max |f(x)| \text{ in the ellipse.} \quad (5.19)$$

5.2.2. Multiple dimensions

The simplest generalization to multiple dimensions is to take the set of nodes $\{\vec{x}_i\}$ to be the outer tensor product of the Chebyshev nodes of Eq. (5.13) for each dimension,

$$\{\vec{x}_i\} = \{x_{i_1}\} \otimes \cdots \otimes \{x_{i_d}\}, \quad (5.20)$$

with possibly different node count in each dimension, n_i . This corresponds to interpolation via nested application of Eq. (5.11), or via the decomposition

$$\tilde{f}(\vec{x}) = \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} c_{i_1 \dots i_d} T_{i_1}(x_1) \cdots T_{i_d}(x_d). \quad (5.21)$$

A detailed study of the interpolation error of this construction is presented in Ref. [250]. Roughly speaking, it is similar to Eq. (5.18) and Eq. (5.19), except instead of n one must use n_i , which are of the order of $\sqrt[d]{n}$, leading to progressively slower convergence as d increases. This is known as *the curse of dimensionality* [251].

5.2.3. Dimensionally adaptive grid

The tensor product construction is fairly rigid in that it allows for no local refinement; only the per-dimension node counts n_i can be tuned. Such tuning is sometimes referred

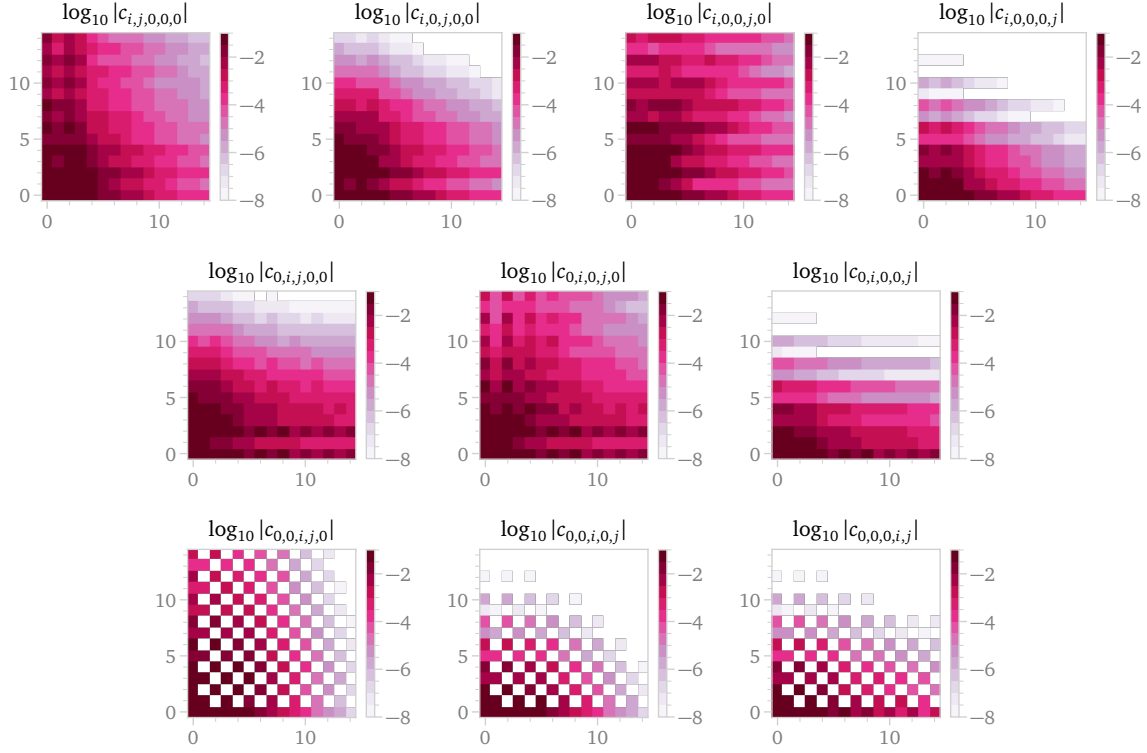


Figure 5.2.: Two-dimensional slices of the five-dimensional coefficients c corresponding to the decomposition of f_1 into a tensor product of Chebyshev polynomials as given in Eq. (5.21). On all plots the horizontal axis is i , the vertical is j .

to as *dimensional adaptivity*, and it can be beneficial. To examine that, let us inspect the coefficients $c_{i_1 \dots i_5}$ corresponding to f_1 : their two-dimensional slices are presented in Fig. 5.2. From these we can learn that the function has discrete symmetries in the 3-rd, 4-th, and 5-th dimensions that make a subset of the basis coefficients zero, forming a checkerboard pattern. We also learn that the coefficients decay much faster in the 5th dimension compared to the rest. To make use of this insight, let us study Fig. 5.3, where the scaling of c is depicted in each dimension. If we want to sample so that coefficients in each direction are close to each other (to prevent oversampling), we would need to maintain a ratio of, e.g.,

$$n_1 : n_2 : n_3 : n_4 : n_5 \approx 2 : 4 : 2 : 3 : 1. \quad (5.22)$$

Better results would of course be obtained if instead of a fixed ratio, we would choose the best n_i ratio for each value of n ; however, this can only be done *a posteriori*.

The results of this optimization, together with the non-dimensionally-adaptive version, are presented in Fig. 5.4. In this figure we show the approximation errors depending on the number of data points n , corresponding to different ways to use polynomial interpolation in practice: different sampling schemes, symmetry handling options, and ways to include weights. The “ a ” method corresponds to interpolating the amplitude a directly, while f corresponds to option (1) from Sec. 5.1.4. The methods marked with “half-domain” correspond to option (iii), while the rest correspond to option (ii).

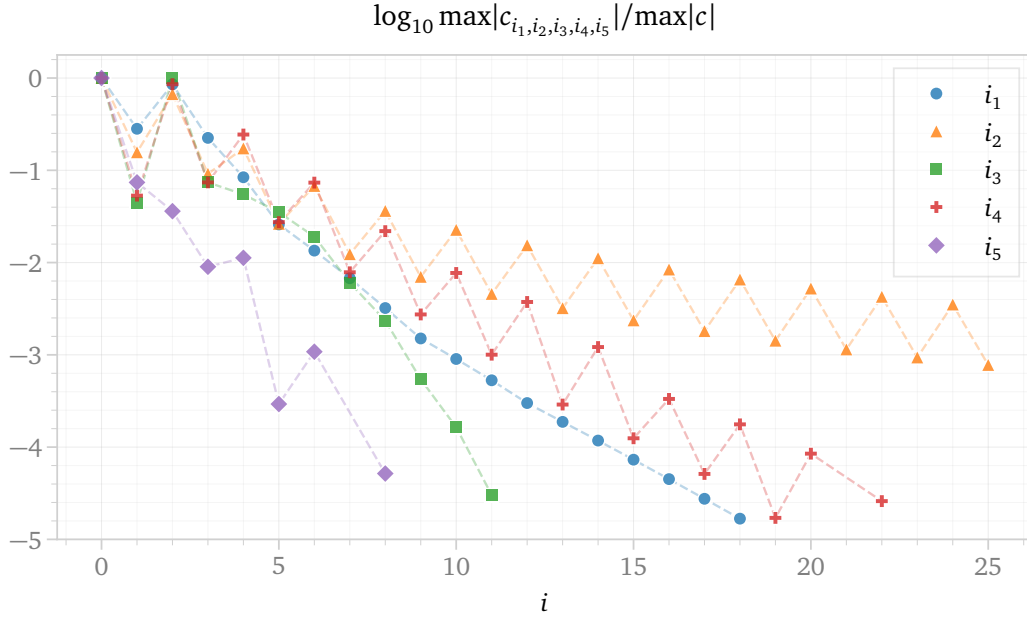


Figure 5.3.: The maximal value of the 5-dimensional coefficients c for f_1 along each of the dimensions, corresponding to the decomposition of Eq. (5.21).

5.2.4. Beyond the full grid

Another way to improve the interpolation is to observe that in Fig. 5.2, even if we optimize the n_i bounds in the cardinal directions, as we did in Sec. 5.2.3, we might still be oversampling in the *diagonal* direction: the right upper parts of each slice show significantly smaller coefficients. The reason is simple: diagonals are longer than the edges [252].

A possible solution is to start with the tensor product basis of Eq. (5.21), but use a different set of multi-indices $\vec{i} \equiv \{i_1, \dots, i_d\}$, such that the upper right corner of Fig. 5.2 is cut out. This will consequentially require choosing a different set of nodes, as we can no longer use the tensor-product construction of Eq. (5.20).

There are many different polynomial bases investigated in the literature (e.g. hyperbolic cross sets [253], lower sets in general [254, 255], including adaptively selected ones [256], sparse sets [257]), and many possible choices for the non-tensor-product nodes (e.g. the specific sets like Morrow–Patterson–Xu points [258, 259], Padua points [260, 261], Chebyshev lattices [262], Lissajous–Chebyshev nodes [263, 264], and general constructions for selecting nodes like Fekete points and Leja sequences [265]). Since we cannot hope to faithfully benchmark every possible combination of options out there, we settle for one instance of this general theme as described in Ref. [266] using the implementation from Ref. [267].

The method consists of constructing the polynomial basis by performing an L^p truncation of the multiindex set in Eq. (5.21), i.e., using all \vec{i} where $\|\vec{i}\|_p \leq k$, where $\|\vec{i}\|_p$ is the L^p vector norm, $(\sum_i i_i^p)^{1/p}$, and choosing the nodes to be the first n points of the tensor-product set of Eq. (5.20) in the Leja order, i.e., chosen one by one greedily, so that the next selected x_i would maximize $\prod_{j < i} (x_i - x_j)$. Results of this method depending on the chosen p are shown in Fig. 5.5.

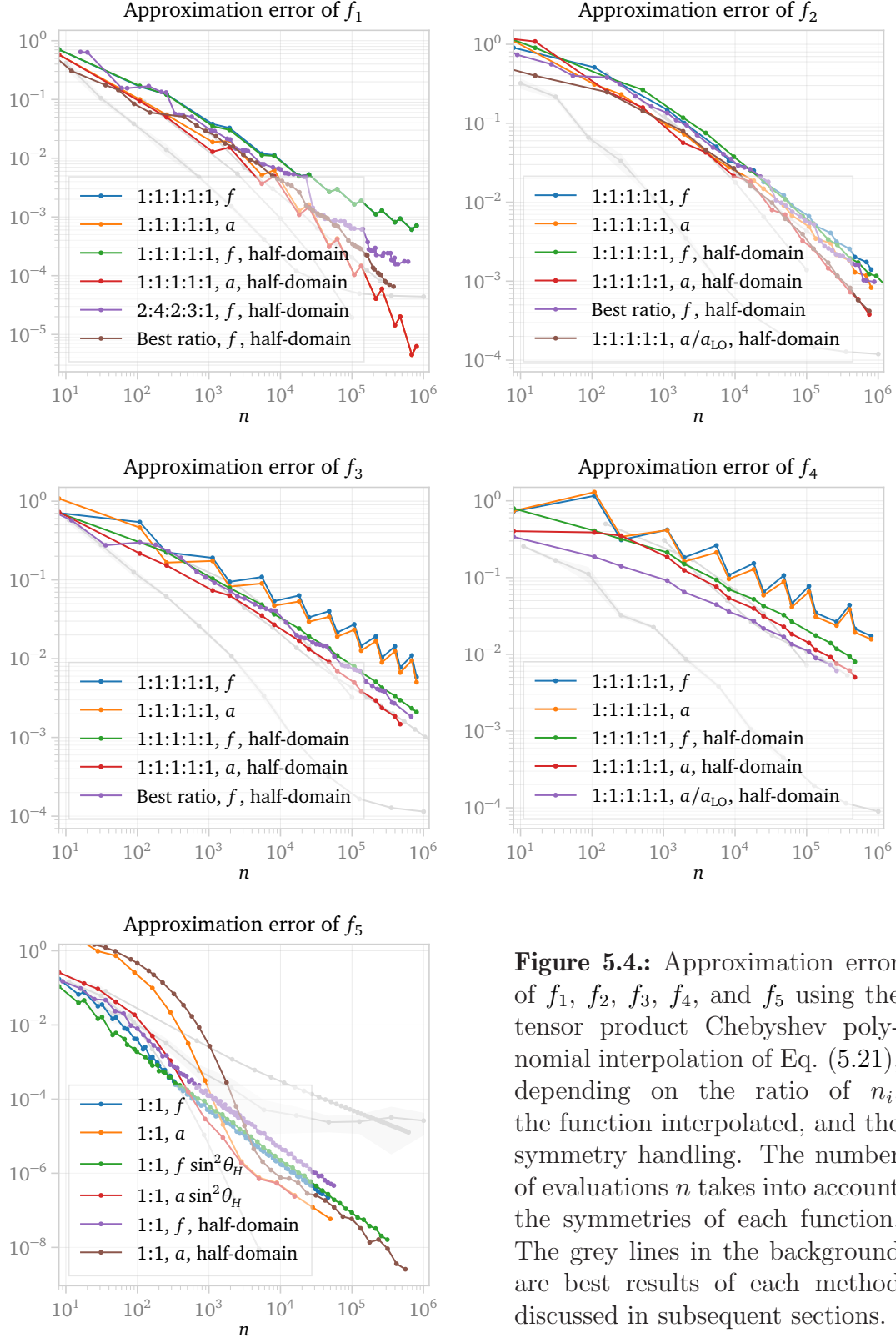


Figure 5.4.: Approximation error of f_1, f_2, f_3, f_4 , and f_5 using the tensor product Chebyshev polynomial interpolation of Eq. (5.21), depending on the ratio of n_i , the function interpolated, and the symmetry handling. The number of evaluations n takes into account the symmetries of each function. The grey lines in the background are best results of each method discussed in subsequent sections.

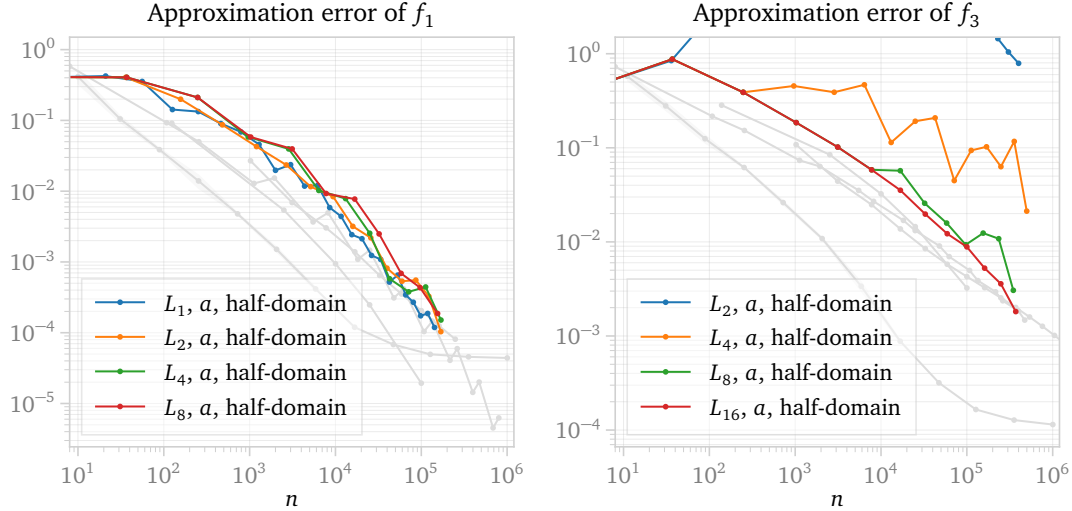


Figure 5.5.: Approximation error (see Eq. (5.2)) of f_1 and f_3 as a function of the number of training points n , using the Leja-ordered non-tensor-product interpolation based on the L^p index set truncation. The “ a ” and “half-domain” labels are the same as in Fig. 5.4.

5.2.5. Effects of noisy data

As a separate important point, we wish to note that the evaluation of $a(\vec{x})$ at any given \vec{x} typically requires more time the higher the precision we demand on it. Thus, it is of practical importance to set a precision target for the evaluations of a . Ideally, this should be done such that it does not increase the approximation error. To study how the approximation error is affected by using low-precision values of a , we can apply multiplicative noise to it:

$$a_i \rightarrow a_i \kappa_i, \quad \kappa \sim \mathcal{N}(1, \sigma^2). \quad (5.23)$$

The result for different σ is shown in Fig. 5.6, where we have intentionally selected very high noise levels to make the effects more visible.

5.2.6. Discussion

Polynomial interpolation is a well understood interpolation method. It achieves exponential convergence rates for analytic functions in one dimension, and is easy to evaluate accurately via the barycentric formula. It should be considered a safe and dependable default method.

Its performance in multiple dimensions, however, is held back by the tensor product grid construction, that inherently comes with the curse of dimensionality. It is also sensitive to singularities close to the interpolation space: the closer the singularity, the worse the convergence becomes—this is particularly inconvenient for amplitudes, which are very often not analytic at boundaries. Finally, the method is inflexible: the node sets that do not suffer from the Runge phenomenon are quite rigid, and if an interpolant was constructed with n data points, one can not easily add just a few more—the best one can do is to use a nested node set, as in Sec. 5.2, and roughly double n . Similarly, if an interpolant was constructed on a subset of the full phase space, there is no good way to smoothly extend it to the rest of the phase space by adding new data.

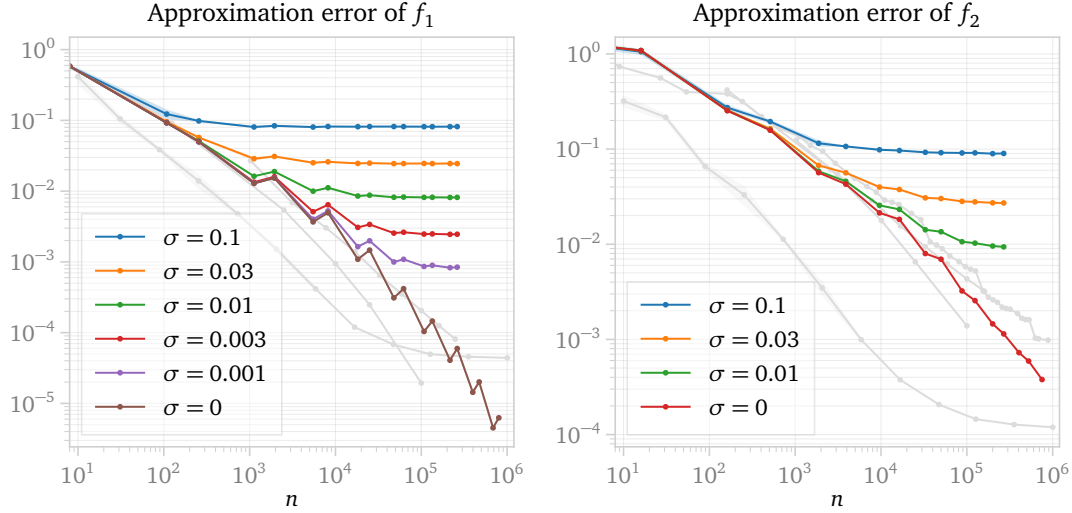


Figure 5.6.: Approximation error (see Eq. (5.2)) of f_1 and f_2 as a function of the number of training points n , if the amplitudes used for evaluation are imprecise, and contain random noise as in Eq. (5.23). The interpolation method corresponds to $1:1:1:1$, a , $half$ -domain from Fig. 5.4.

From the gathered results we can provide insight into the following questions:

- *Should the interpolant be constructed on a or f ?* Including the weight w (i.e., option (1) from Sec. 5.1.4) helps a lot for very peaky functions (f_5), but seems to slightly hinder others. A possible reason is that w is not analytic at the boundaries, which is a sensitive environment for polynomial interpolation.
- *Does taking a ratio to the leading order help?* Since f_1 is the leading-order version of f_2 , and f_3 of f_4 , it is natural to ask if interpolating f_2/f_1 (the so-called K -factor) is better than directly f_2 or a_2 . For f_2 this appears to make no difference, while for f_4 there is a notable improvement at lower precisions.
- *How should the symmetries be included?* Domain reduction (i.e., option (iii) from Sec. 5.1.4) significantly helps f_3 and f_4 , makes almost no difference for f_1 and f_2 , and slightly hinders f_5 , for which data duplication (i.e., option (ii)) is better.
- *Does factoring out the known peaky behaviour from the interpolant help?* Yes, cancelling the $1/\sin^2 \theta_H$ factor from the f_5 interpolant brings a modest but notable improvement. Subtracting the peak could have been even better though.
- *Does dimensional adaptivity help?* It only makes a notable improvement for f_1 .
- *Does non-full-grid sampling help?* Not the version of it that we investigated.
- *How precise must the data points be?* The noise magnitude σ acts as a lower boundary for ε , so σ must be chosen at least as $\sigma \leq \varepsilon_{\text{target}}$. Values of $\sigma \lesssim 1/3 \varepsilon_{\text{target}}$ seem to allow reaching the desired ε while being virtually unaffected by noise.

5.3. B-spline interpolation

Next, we move on to consider a close relative of global polynomials, the B-splines [268, 269]. A *spline* of degree p is a $p - 1$ times continuously differentiable *piecewise* polynomial. Splines provide an alternative way of preventing the Runge phenomenon by limiting the degree of the polynomial, see Eq. (5.12). We use in particular B-splines (basis splines), since they are the standard basis choice for spline interpolation.

B-splines have been applied in engineering applications in the context of the finite element method [270] and in graphics as *non-uniform rational B-splines* (NURBS) [271]. B-splines have also been successfully used for interpolating amplitudes in lower-dimensional cases [272, 273], which makes this a particularly interesting method for us to compare with. Spline interpolation has also been applied in the context of PDF fits [274].

In this section we define the B-spline basis functions and show how to define multidimensional B-spline interpolants on tensor product grids. We then present the resulting approximation errors from using uniform B-splines and finish with a discussion of the obtained performance.

5.3.1. B-spline basis functions

A B-spline basis function of degree p and index i is defined recursively through the Cox-de Boor formula [275, 276]:

$$\begin{aligned} N_{i,0}(x) &= \begin{cases} 1 & \text{if } t_i \leq x \leq t_{i+1}, \\ 0 & \text{otherwise,} \end{cases} \\ N_{i,p}(x) &= \frac{x - t_i}{t_{i+p} - t_i} N_{i,p-1}(x) + \frac{t_{i+p+1} - x}{t_{i+p+1} - t_{i+1}} N_{i+1,p-1}(x), \end{aligned} \quad (5.24)$$

where t_i are the coordinates of *knots*—the locations at which the piecewise functions meet.

Knots define a *knot vector* \vec{t} , which is a sequence of $m + 1$ non-decreasing numbers (t_0, \dots, t_m) . The number of knots is related to the degree through $m = n + p + 1$, where $n + 1$ is the number of basis functions [277]. In an interpolation context, the first and last $p + 1$ knots should be placed outside or at the boundary of the interpolation domain, such that the lower and upper boundaries correspond to t_p and t_{m-p} . Under these conditions the *local Marsden identity* is fulfilled, which ensures that the B-spline basis represents polynomials exactly on the domain (t_p, t_{m-p}) [278, 279]. This ensures that the B-spline has approximation power across the entire phase space. The most straightforward choice of knot vector is therefore the uniform construction, with p auxiliary knots placed outside the interpolation domain. The basis functions resulting from such uniform knot vectors for $p = 1, 2, 3$ are shown in Fig. 5.7.

Another common choice is the *not-a-knot* construction, where knots coincide with data points, except at $p - 1$ points, which are omitted. Since in the even degree case this results in an odd number of not-a-knots, this knot vector is better defined for odd degree B-splines. Note that the linear case simplifies to the uniform construction since there are 0 not-a-knots in this case.

From Eq. (5.24) it can be seen that the basis functions are non-negative, form a partition of unity and have local support. Moreover, thanks to the recursive nature of

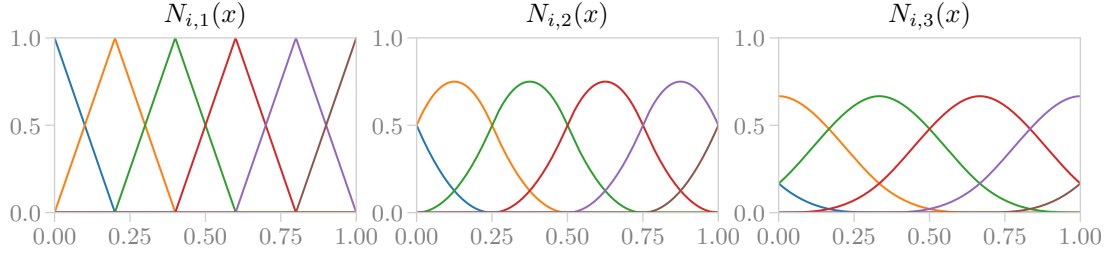


Figure 5.7.: Linear, quadratic and cubic B-spline basis functions from Eq. (5.24) with uniform knot vectors, for different i .

Eq. (5.24), there are efficient algorithms that make B-splines fast to evaluate compared to other spline functions [277]. This is especially true for B-splines with uniform knot vectors, since the denominators in Eq. (5.24) are constant. In the not-a-knot case the knot distances are not all equal, but can be precomputed in advance.

5.3.2. B-spline interpolants

A one-dimensional B-spline interpolant is a linear combination of $n + 1$ basis functions

$$b(x) = \sum_{i=0}^n N_{i,p}(x) \cdot c_i, \quad (5.25)$$

where the c_i are interpolation coefficients sometimes referred to as *control points*. The most straightforward extension to the d -dimensional case is through a tensor product construction

$$B(\vec{x}) = \prod_{j=1}^d b_j(\vec{x}_j). \quad (5.26)$$

To fully constrain a B-spline with $n + 1$ basis functions in each direction, $(n + 1)^d$ interpolation nodes are required. The interpolation nodes can partially be selected freely but need to satisfy the Schoenberg–Whitney conditions, which are degree dependent and state that for each knot t_i there must be at least one data point x such that $t_i < x < t_{i+p+1}$. Moreover, the properties of the basis functions imply that a B-spline interpolant is numerically stable [280].

5.3.3. Discussion

In Fig. 5.8 the performance of B-spline interpolation is compared to the other methods described in this chapter. For all test functions a significant performance gain over linear splines is obtained by using basis functions of at least quadratic degree. For f_1 and f_2 cubic splines perform slightly better than quadratic ones; for f_3 , f_4 , and f_5 quadratic splines are the best.

The effect of interpolating either the amplitude (“ a ”) or the test function (“ f ”) is shown for f_1 , f_3 and f_5 . For f_1 and f_3 it is better to directly interpolate the amplitude. For f_5 including the weight helps in the low training-data regime, but becomes worse for high amount of training data.

B-splines suffer from the same flexibility issues as polynomials, as discussed in Sec. 5.2.6, because the extension of the B-splines to the multivariate case is based on the same tensor product construction. It is difficult to predict exactly how many points are required for a certain precision target, so the required number of evaluations likely overshoots the minimal number of evaluations. Therefore, the results in Fig. 5.8 are optimistic from a practical point of view. We describe in Sec. 5.4 how B-splines can be combined with an adaptive approach in the context of sparse grids, which removes this problem.

Additionally, B-splines require function evaluations at the boundaries, which is also pointed out in Sec. 5.2.6 to be a drawback for amplitude interpolation. For example, it is observed in Ref. [16] that the evaluation time diverges close to certain boundaries. For such cases, the boundary of the interpolation space can be shifted to a point where the evaluation time is reasonable, meaning that the approximation would not cover some parts of the phase space. Alternatively, methods that incorporate extrapolation, such as the modified bases on adaptive sparse grids from Sec. 5.4, can be used to alleviate this problem.

Where B-splines shine is in their evaluation speed: a B-spline interpolant can be evaluated in small constant time, whereas a polynomial interpolant needs time proportional to the number of data points. For this reason, B-splines can speed up the evaluation of more precise approximation methods by first constructing the approximation using those methods, and then approximating that approximation using B-splines. This is possible because the data points for the second approximation will be obtained by evaluating the first approximation, which should be much faster than evaluating the target function.

5.4. Sparse grids

The methods described in the previous sections are based on the tensor product construction of Eq. (5.20), which results in a *full grid*. A *sparse grid* construction aims to omit points from the full grid that do not significantly contribute to the interpolant, and in this way alleviate the curse of dimensionality. Such constructions were first described by Smolyak [281], then rediscovered by Zenger [282] and Griebel [283], and have since found use in a wide variety of applications [284–290].

In this section we first introduce sparse grids built on a hierarchical linear basis. Next, we describe how to incorporate spatial adaptivity and upgrade the basis to higher degree polynomials. Finally, we study the impact these constructions have on the approximation quality.

5.4.1. Classical sparse grids

Sparse grids can be constructed in two main ways, either with the *combination technique* using a linear combination of full grids [255], or through a hierarchical decomposition of the approximation space [286]. In this section we use the latter approach, since it makes it straightforward to incorporate spatial adaptivity.

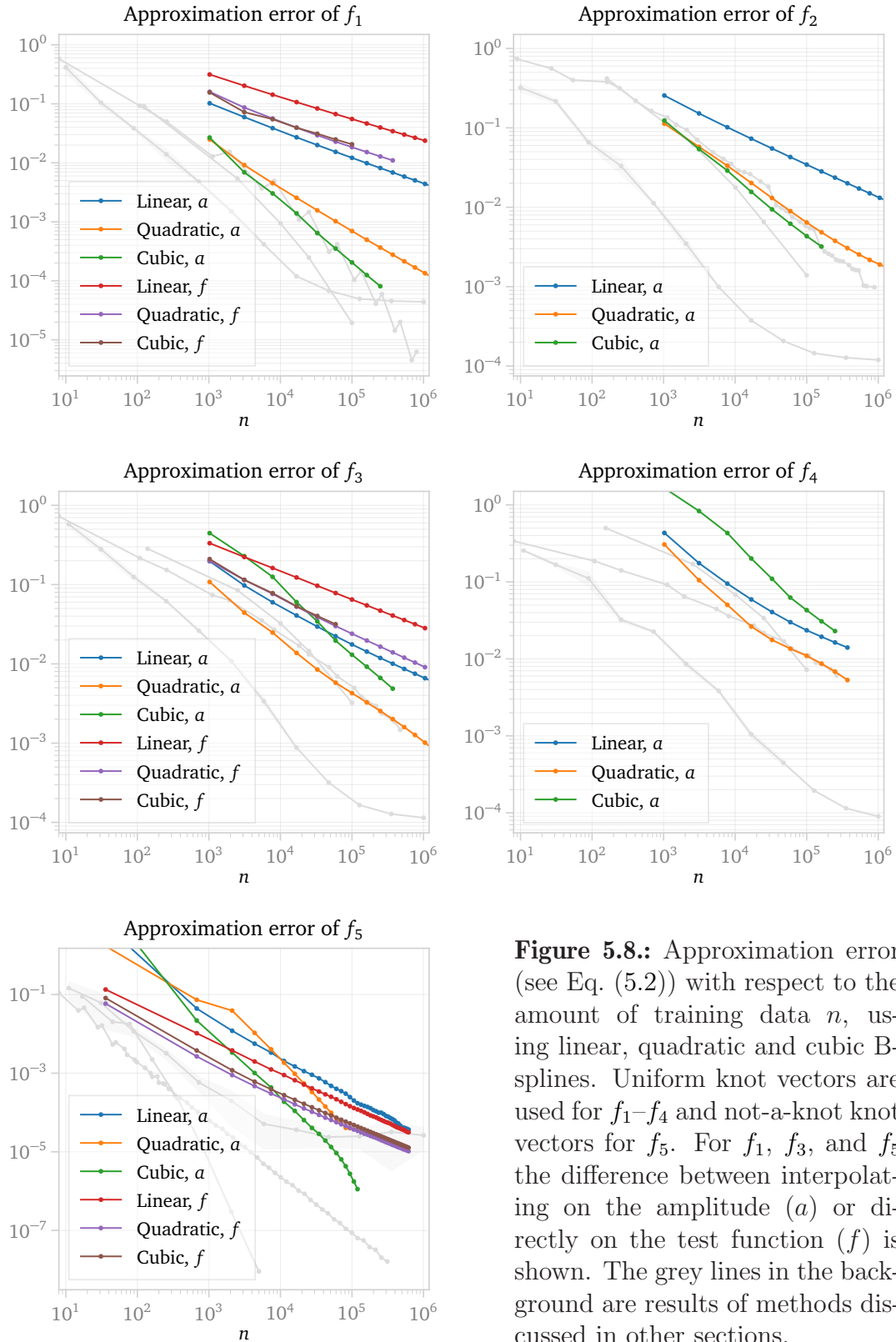


Figure 5.8.: Approximation error (see Eq. (5.2)) with respect to the amount of training data n , using linear, quadratic and cubic B-splines. Uniform knot vectors are used for f_1 – f_4 and not-a-knot knot vectors for f_5 . For f_1 , f_3 , and f_5 the difference between interpolating on the amplitude (a) or directly on the test function (f) is shown. The grey lines in the background are results of methods discussed in other sections.

First, let us restrict to functions that vanish at the boundaries. In this case, a one-dimensional basis can be constructed with rescaled “hat” functions that are centred around the grid points $x_{l,i} = i \cdot 2^{-l}$, $i \in \{0, 1, \dots, 2^l\}$:

$$\phi_{l,i}(x) \equiv \phi\left(\frac{x - i \cdot 2^{-l}}{2^{-l}}\right), \quad \phi(x) \equiv \max(1 - |x|, 0), \quad (5.27)$$

where l is the *grid level*. Since these basis functions have local support, it is possible to define *hierarchical subspaces* W_l through the *hierarchical index* sets I_l by

$$W_l \equiv \text{span}\{\phi_{l,i} : i \in I_l\}, \quad I_l \equiv \{i \in \mathbb{N} : 1 \leq i \leq 2^l - 1, i \text{ is odd}\}. \quad (5.28)$$

The function space of one-dimensional interpolants is then defined as the direct sum of all subspaces up to a maximum grid level k

$$V_k \equiv \bigoplus_{l \leq k} W_l. \quad (5.29)$$

The extension to the multivariate case is via a tensor product construction

$$\Phi_{\vec{l}, \vec{i}}(\vec{x}) \equiv \prod_{j=1}^d \phi_{l_j, i_j}(x_j), \quad (5.30)$$

with multi-indices $\vec{i} = (i_1, \dots, i_d)$, $\vec{l} = (l_1, \dots, l_d)$ and d -dimensional grid points $\vec{x}_{\vec{i}, \vec{l}} = (x_{l_1, i_1}, \dots, x_{l_d, i_d})$. Similarly to the one-dimensional case, the hierarchical subspaces are defined to be

$$W_{\vec{l}} \equiv \text{span}\{\Phi_{\vec{l}, \vec{i}} : \vec{i} \in I_{\vec{l}}\}, \quad I_{\vec{l}} \equiv \{\vec{i} : 1 \leq i_t \leq 2^{l_t} - 1, i_t \text{ is odd}, 1 \leq t \leq d\}. \quad (5.31)$$

A multidimensional full-grid function space V_k^F can now be naively constructed with the direct sum

$$V_k^F \equiv \bigoplus_{|\vec{l}|_\infty \leq k} W_{\vec{l}}, \quad (5.32)$$

where $|\vec{l}|_\infty = \max(l_1, \dots, l_d)$. The mechanism of the sparse grid is to instead limit the selection of subspaces according to

$$V_k^S \equiv \bigoplus_{|\vec{l}|_1 \leq k+d-1} W_{\vec{l}}, \quad (5.33)$$

where $|\vec{l}|_1 = l_1 + \dots + l_d$. This avoids including basis functions that are highly refined in all directions simultaneously, which is what causes the number of grid points to explode in high dimensional cases.

Fig. 5.9 shows nodes of the resulting two-dimensional sparse grids for $k = 2, 3, 4$. Table 5.1 shows the difference in the number of grid points between sparse and full constructions, for dimensions 2 and 5. Of course, omitting grid points can never increase the approximation quality, but the aim is to achieve a close approximation quality, while omitting most points. For sufficiently smooth functions, it is known that the asymptotic accuracy of a full grid interpolant scales with the mesh-width $h_k = 2^{-k}$ as $\mathcal{O}(h_k^2)$, while for a sparse grid it decreases only by a logarithmic factor to $\mathcal{O}(h_k^2 (\log h_k^{-1})^{d-1})$ (see Ref. [286]).

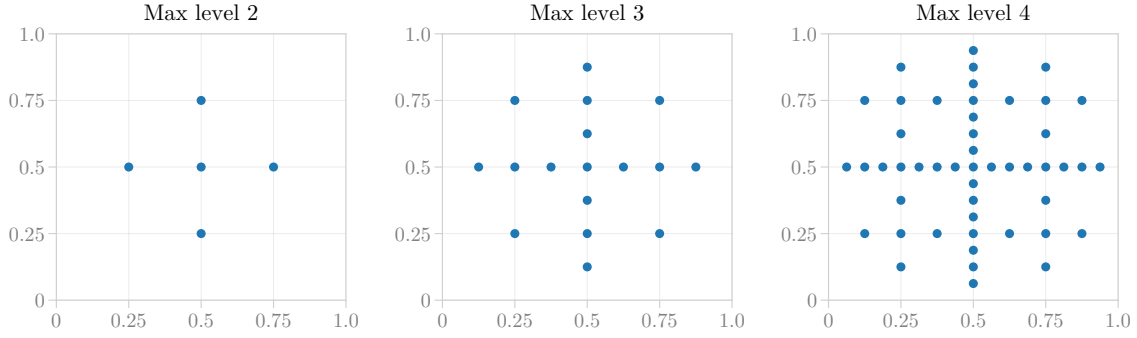


Figure 5.9.: Sparse grid structure in two dimensions for $k = 2, 3, 4$.

Grid type	$d \setminus k$	2	4	6	8	10
Full	$d = 2$	25	289	4225	$6.6 \cdot 10^4$	$1.1 \cdot 10^6$
	$d = 5$	3125	$1.4 \cdot 10^6$	$1.2 \cdot 10^9$	$1.1 \cdot 10^{12}$	$1.1 \cdot 10^{15}$
Sparse	$d = 2$	5	49	321	4097	$9.2 \cdot 10^3$
	$d = 5$	11	315	5503	$6.1 \cdot 10^4$	$5.5 \cdot 10^5$

Table 5.1.: Number of grid points from the full and sparse constructions, for dimensions 2 and 5.

5.4.2. Boundary treatment

Since the basis functions of Eq. (5.27) do not have support at the grid boundaries, we can, so far, only handle target functions that vanish at the boundary. There are two main ways to incorporate boundary support into sparse grids. One possibility is to add boundary points to the grid [286], but for even moderately high dimension this causes the number of points to increase significantly, and defeats the main purpose of the sparse construction. The other option is to use so-called *modified* basis functions [286] that linearly extrapolate from the outermost grid points:

$$\varphi_{l,i}(x) \equiv \begin{cases} 1 & \text{if } l = 1 \wedge i = 1, \\ \begin{cases} 2 - 2^l \cdot x & \text{if } [0, \frac{1}{2^{l-1}}], \\ 0 & \text{otherwise.} \end{cases} & \text{if } l > 1 \wedge i = 1, \\ \begin{cases} 2^l \cdot x + 1 - i & \text{if } x \in [1 - \frac{1}{2^{l-1}}, 1], \\ 0 & \text{otherwise.} \end{cases} & \text{if } l > 1 \wedge i = 2^l - 1, \\ \phi_{l,i}(x) & \text{otherwise.} \end{cases} \quad (5.34)$$

Fig. 5.10 shows the modified linear basis for $k = 1, 2, 3$. It is apparent that sparse grids are best suited for functions whose boundary behaviour is less important than that in the bulk to the overall structure. The interpolant is constructed as a linear combination of basis functions in V_k^S according to

$$u(\vec{x}) = \sum_{|\vec{l}|_1 \leq k+d-1} \sum_{\vec{i} \in I_{\vec{l}}} \alpha_{\vec{l},\vec{i}} \cdot \Phi_{\vec{l},\vec{i}}(\vec{x}), \quad (5.35)$$

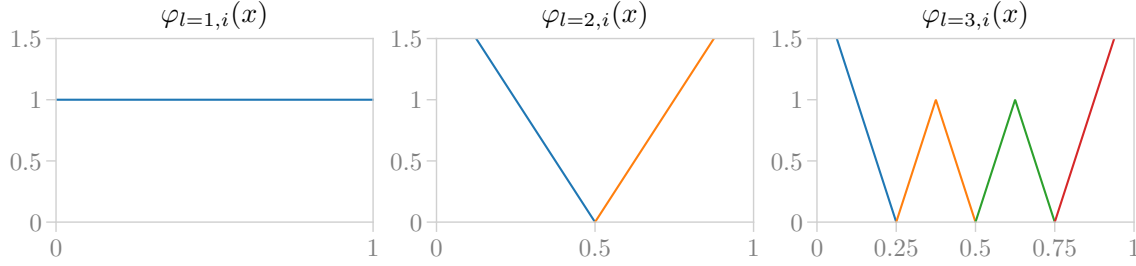


Figure 5.10.: Modified linear hat-basis for sparse grid levels $l = 1, 2, 3$ and different i .

where the interpolation coefficients $\alpha_{\vec{l},\vec{i}}$ are referred to as *hierarchical surpluses* since at each level they correct the interpolant from the previous level to the target function. They are thus also a measure of the absolute error at each level in each direction, which makes the interpolant $u(\vec{x})$ well suited for local adaptivity. The basis functions $\Phi_{\vec{l},\vec{i}}(\vec{x})$ are constructed from either Eq. (5.27) or Eq. (5.34) depending on if the target function vanishes at the boundary or not. The process of determining the coefficients $\alpha_{\vec{l},\vec{i}}$ is known as *hierarchization*. This can in principle be done in the same way as for any interpolation method; construct and solve a linear system of equations where the interpolant is demanded to reproduce the target function at each interpolation node. A more efficient way, however, is available for bases satisfying the *fundamental* property:

$$\varphi_{l,i}(2^{-l}i) = 1 \quad \text{and} \quad \varphi_{l,i}(2^{-l}(i-1)) = \varphi_{l,i}(2^{-l}(i+1)) = 0. \quad (5.36)$$

In this case the coefficients can be calculated on the fly: the grid is initialized by normalizing the first coefficient to the central value $\alpha_{\vec{1},\vec{1}} = f(0.5, \dots, 0.5)$; points $\vec{x}_{\vec{l},\vec{i}}$ are then added, one at a time, and the coefficients are defined as the corrections

$$\alpha_{\vec{l},\vec{i}} = f(\vec{x}_{\vec{l},\vec{i}}) - \tilde{f}(\vec{x}_{\vec{l},\vec{i}}), \quad (5.37)$$

where $\tilde{f}(\vec{x}_{\vec{l},\vec{i}})$ is the current approximation with the already added points and $f(\vec{x}_{\vec{l},\vec{i}})$ is the true function value.

5.4.3. Spatially adaptive sparse grids

While the classical sparse grid construction uses significantly fewer points than a full grid, it is completely blind to how the target function varies across the parameter space. In many high dimensional problems, it is typical that some regions are more important than others. This information is exploited by spatially adaptive sparse grids, where the grid points that contribute most to the interpolant are *refined* first (see Refs. [286, 291]). Refining a grid point means adding all its neighbouring points at one level lower to the grid.

Determining which points are more important relies on some heuristic criterion. A common strategy is to use the *surplus criterion*, where the point with the largest hierarchical surplus is refined first. This is the *greedy strategy*. Its disadvantage is that it might get stuck refining small regions of the parameter space. To avoid this, in Ref. [286] it is proposed to weigh the surplus criterion by the volume of the support of the corresponding basis function, $2^{-|\vec{l}|_1}$, resulting in the *balanced strategy*.

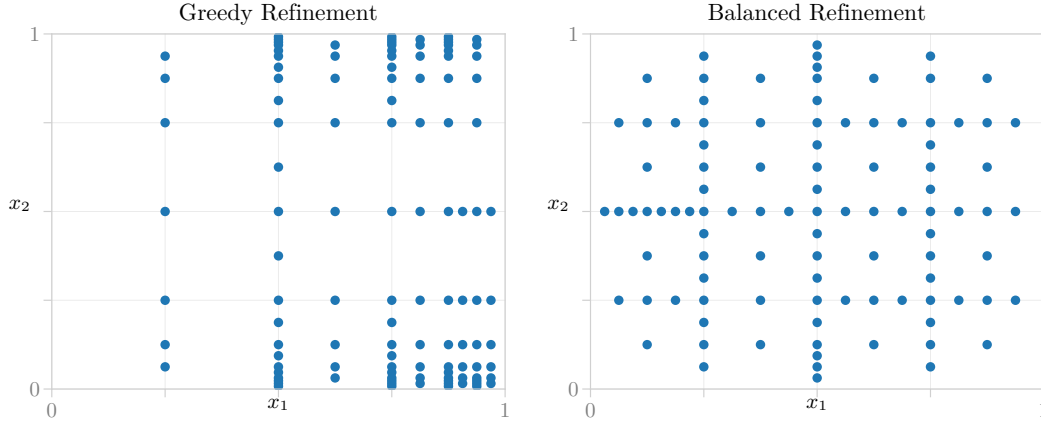


Figure 5.11.: Point distribution from spatial adaptivity with greedy and balanced refinement for the two-dimensional test function f_5 .

Fig. 5.11 shows the difference between greedy and balanced refinement on a two-dimensional grid, for the test function f_5 . We find that greedy refinement might perform slightly better for our error metric if the structure of the target function is simple enough, while balanced refinement is significantly more reliable for more complicated target functions.

5.4.4. Higher degree basis functions

The one-dimensional basis of rescaled hat functions in Eq. (5.27) is straightforward to implement and avoids the Runge phenomenon due to its piecewise nature. The drawback is that the linear approximations come with low interpolation performance, as we have previously observed in Sec. 5.3.

There are many ways to make the extension to higher polynomial degrees. Here we differentiate between two types of basis functions: those that fulfil the fundamental property Eq. (5.36) and those that do not.

Fundamental basis functions allow for fast hierarchization and make it easy to construct efficient evaluation algorithms. In addition to the linear basis from Sec. 5.4.1 we test the more general polynomial piecewise functions (C0-elements) [286, 292], as well as fundamental B-splines [289]. For these benchmarks we make use of the sparse grid toolbox SG++ [286], where both of these basis functions are already implemented. Beyond fundamental basis functions we also investigate extended not-a-knot B-splines as described in Refs. [290, 293].

C0 elements

Higher degree polynomials can be defined on the hierarchical structure by using grid points on upper levels as the polynomial nodes [286, 292]. This implies the maximum degree p is bounded by the maximum grid level. For sparse grids without boundaries the relation is $p \leq l - 1$. This can be seen already for the linear case in Fig. 5.10, since on levels 1 and 2 the basis functions are constant and linear respectively. The drawback of this extension is that despite the higher degree, the smoothness is not increased and the interpolant will only be continuous. For this reason these basis functions are usually referred to as C0 elements.

p	$e_{i,j=0}, \quad i = 1, \dots, p+1$	$e_{i,j=2^l}, \quad i = 2^l - 1 - p, \dots, 2^l - 1$
1	$[2, -1]$	$[-1, 2]$
3	$[5, -10, 10, -4]$	$[-4, 10, -10, 5]$
5	$\begin{cases} [8, -28, 42, -35, 20, -6] & \text{if } l = 3, \\ [8, -28, 56, -70, 56, -21] & \text{if } l > 3. \end{cases}$	$\begin{cases} [-6, 20, -35, 42, -28, 8] & \text{if } l = 3, \\ [-21, 56, -70, 56, -28, 8] & \text{if } l > 3. \end{cases}$

Table 5.2.: B-spline extension coefficients for the upper and lower boundaries, taken from Ref. [293].

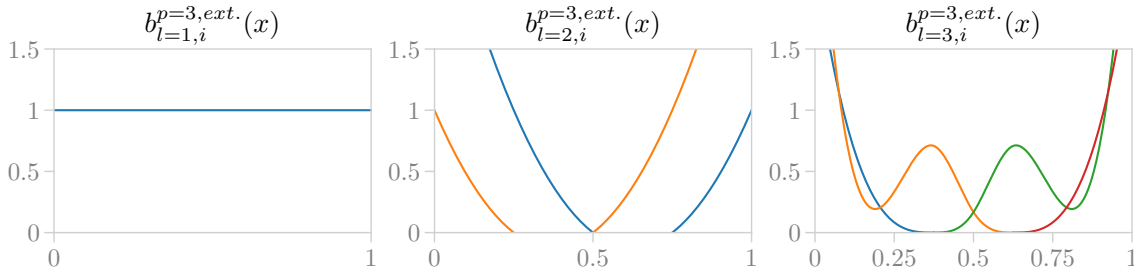


Figure 5.12.: Extended not-a-knot cubic B-spline basis functions at sparse grid levels $l = 1, 2, 3$ and different i . At level 1 there are not enough points to define the B-spline and a linear Lagrange polynomial is used instead.

Fundamental B-splines

The fundamental B-spline basis is constructed with the aim of fulfilling the fundamental property of Eq. (5.36), while preserving useful properties of B-splines, for example smoothness. The construction is introduced in Ref. [289], and it works by applying a translation-invariant fundamental transformation to the hierarchical B-spline basis. Besides fulfilling the fundamental property, this transformation preserves the translational invariance of B-splines which improves performance during evaluation. The modified basis that extrapolates toward the boundaries is defined similarly to the linear case. We refer to Ref. [289] for more details on the derivation. For both this and the C0 elements, the implementations in SG++ are used for the benchmarks.

Extended not-a-knot B-splines

In this section we summarize the main equations and statements on extended not-a-knot B-splines presented in Refs. [290, 293]. The extension mechanism ensures that polynomials are interpolated exactly, which in many cases increases the quality of interpolation. The basis consists of extended not-a-knot B-splines on lower levels, and Lagrange polynomials on upper levels where there are too few points for the B-spline to be defined. A basis function of odd degree p , level l and index i is defined as

$$b_{l,i}^{p,\text{ext.}}(x) = \begin{cases} b_{l,i}^p + \sum_{j \in J_l(i)} e_{i,j} b_{l,j}^p & \text{when } l \geq \Lambda^{\text{ext.}}, \\ L_{l,i}(x) & \text{when } l < \Lambda^{\text{ext.}}, \end{cases} \quad (5.38)$$

where $\Lambda^{\text{ext.}} = \lceil \log_2(p+2) \rceil$, and $e_{i,j}$ are extension coefficients that only depend on the degree. For $p = 1, 3, 5$ these coefficients are listed in Table 5.2. The $b_{l,i}^p$ are not-a-knot B-splines (see Sec. 5.3) and $L_{l,i}(x)$ are regular Lagrange polynomials defined with a uniform knot vector $t = (0, 2^{-l}, \dots, 1)$ as

$$L_{l,i}(x) = \prod_{\substack{1 \leq m \leq 2^l - 1, \\ m \neq i}} \frac{x - t_m}{t_i - t_m}, \quad i = 0, \dots, 2^l. \quad (5.39)$$

The index set $J_l(i)$ defines the extension of the B-spline. It determines to which interior basis functions the boundary basis functions are added. An interior basis function of index i includes the boundary basis function if i is among the first or last $p+1$ interior indices. Formally the index set is defined as $J_l(i) = \{j \in J_l \mid i \in I_l(j)\}$, where $J_l = \{0, 2^l\}$ and $I_l(0) = \{1, \dots, p+1\}$, $I_l(2^l) = \{2^l - 1 - p, \dots, 2^l - 1\}$ (see Ref. [293]). The basis functions described by Eq. (5.38) are shown in Fig. 5.12 for the cubic case. The linear case simplifies to the linear hat-basis described in Sec. 5.4.1.

5.4.5. Discussion

First, we compare the performance of different refinement criteria in Fig. 5.13 for test functions f_1 , f_3 , and f_5 , using the modified linear basis. In some cases the greedy construction is slightly better, but for example the results for f_3 demonstrate the danger of this refinement strategy. The balanced construction results in a much smaller approximation error, which hints at the greedy algorithm getting stuck refining local structures. In practice, it is difficult to predict when refining greedily is better, and in such cases the advantage seems to not be very significant. Moreover, greedy refinement is sensitive to noise in the training data due to random enhancement of hierarchical surpluses. Finally, balanced refinement is more robust against noise since it suppresses the effect of local deviations by construction. For these reasons, balanced refinement is the more reliable approach for our studies. We also try to weigh the refinement criterion by the phase-space weight w from Eq. (5.8). For f_1 and f_3 this results in a more uniform target function, and the resulting sparse grid becomes similar to the balanced construction without weighting. We therefore see only minor differences in the results from balanced and weighted refinement.

In Fig. 5.14 we compare the approximation error from spatially adaptive sparse grids constructed with balanced refinement against the other methods described in this chapter. With other basis choices we see significant improvements for all test functions. In particular when increasing the degree with the piecewise polynomial and fundamental B-spline bases. With the extended not-a-knot B-splines we see an even better improvement for f_5 , but for f_1 – f_4 the performance is at best similar to the linear case. In particular for f_3 and f_4 the cubic and quintic cases converge very poorly. This result is unexpected since these bases have been applied to high-dimensional test functions with good results in Ref. [293]. As a cross-check, we have reproduced those benchmarks with our implementation and our results with SG++.

For the loop amplitudes f_2 and f_4 we also try to flatten the target function by interpolating their ratios to the corresponding leading-order amplitudes (f_1 and f_3). Fig. 5.15 shows the results for the linear, polynomial and fundamental spline bases. We see a significant improvement in the low-data regime, but the scaling is worse compared to

interpolating the amplitude directly. Additionally, the higher degree basis functions appear to be more robust against the worse scaling.

The improvement of the spatially adaptive sparse grids over the non-adaptive full grid methods at low amounts of training data is modest. However, we observe a better scaling overall for each test function, yielding a significant performance boost at high data regimes. Aside from interpolation performance, a major advantage of spatially adaptive sparse grids is flexibility. Since it is difficult to predict how much training data is required to reach a certain precision target for an unknown function, it is likely that training data needs to be added iteratively. As is discussed in Sec. 5.2 and Sec. 5.3, non-adaptive methods are very limited in this regard. A spatially adaptive sparse grid on the other hand is able to add one point at a time, making it possible to validate during construction and in principle stop at exactly the required number of points. If a non-adaptive method is used to reconstruct an unknown function, we are likely to overshoot the required number of points, making the effective number of function evaluations higher than what is represented by these results. The results for the non-adaptive methods at a given training size are therefore the best case scenario, while the sparse grid results are close to what one obtains in practice.

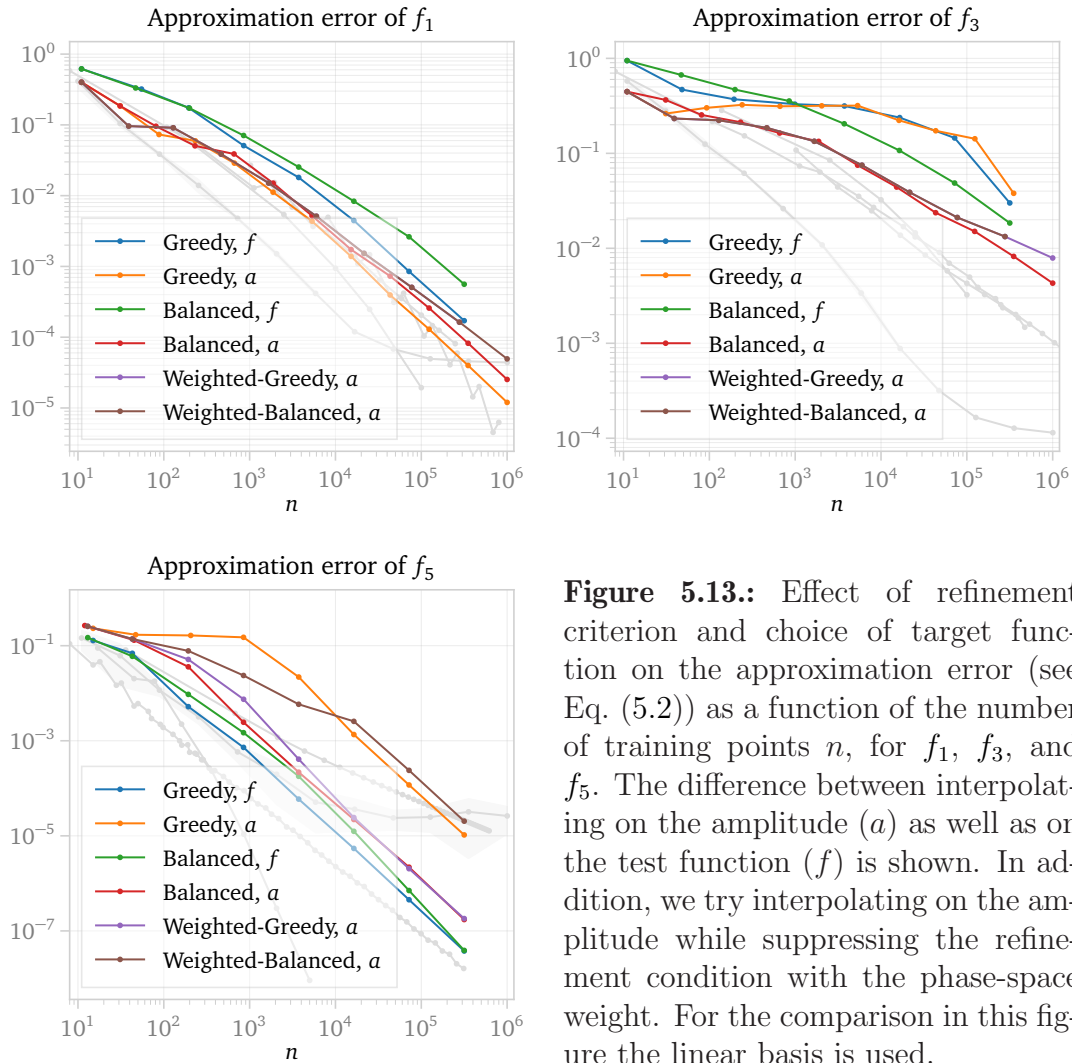


Figure 5.13.: Effect of refinement criterion and choice of target function on the approximation error (see Eq. (5.2)) as a function of the number of training points n , for f_1 , f_3 , and f_5 . The difference between interpolating on the amplitude (a) as well as on the test function (f) is shown. In addition, we try interpolating on the amplitude while suppressing the refinement condition with the phase-space weight. For the comparison in this figure the linear basis is used.

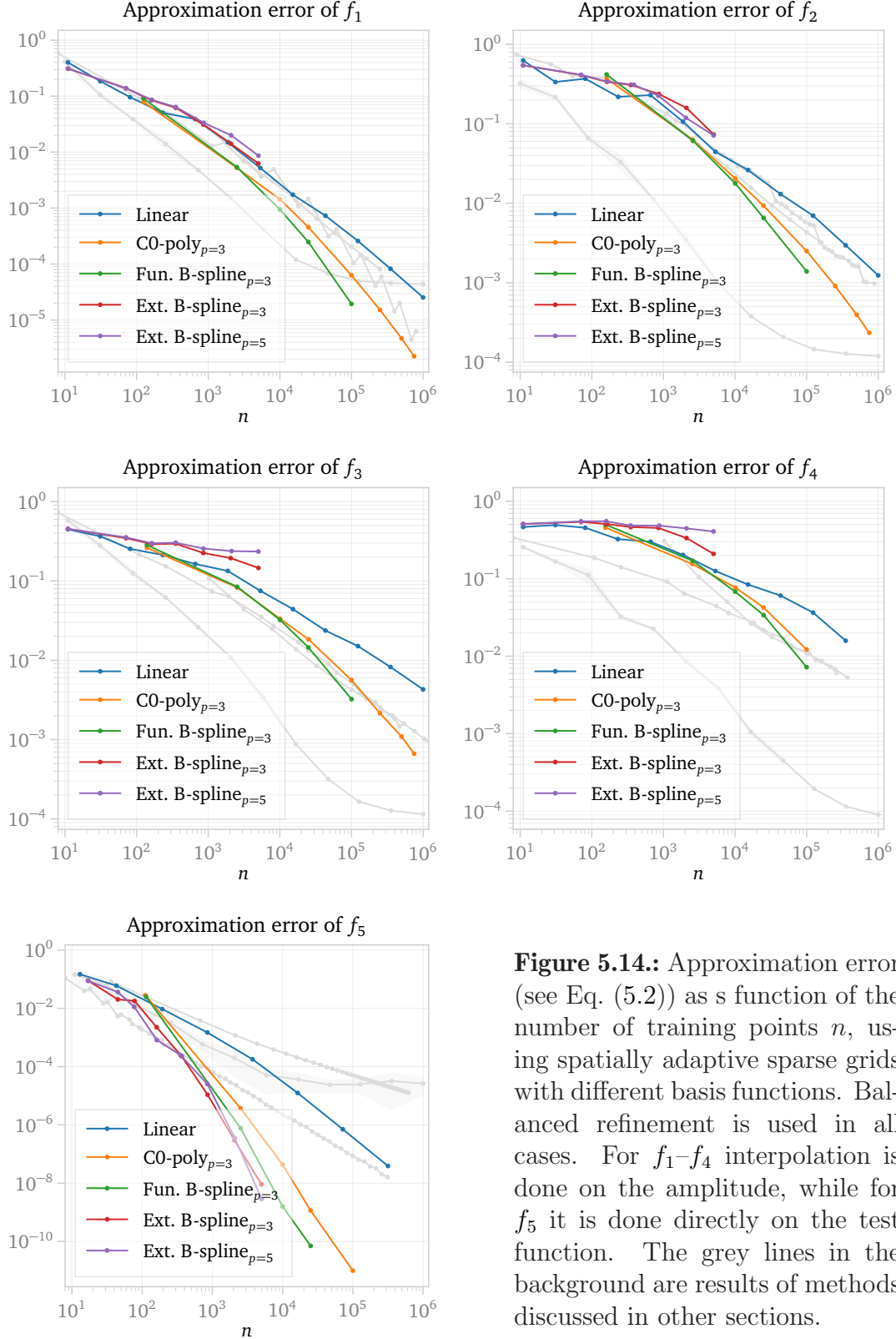


Figure 5.14.: Approximation error (see Eq. (5.2)) as a function of the number of training points n , using spatially adaptive sparse grids with different basis functions. Balanced refinement is used in all cases. For f_1 – f_4 interpolation is done on the amplitude, while for f_5 it is done directly on the test function. The grey lines in the background are results of methods discussed in other sections.

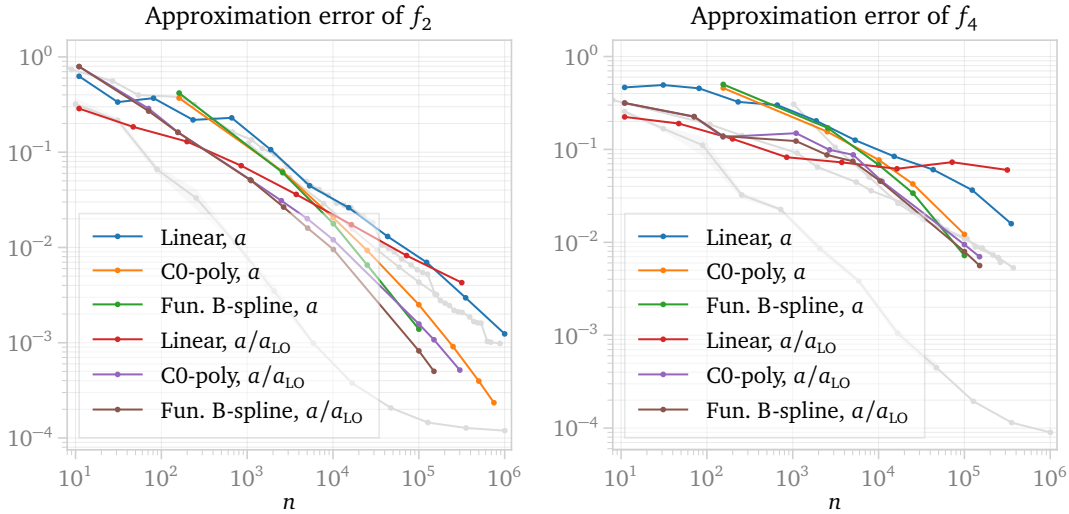


Figure 5.15.: Approximation error (see Eq. (5.2)) as a function of the number of training points n , using sparse grids with balanced refinement targeting amplitudes a and the ratios a/a_{LO} respectively. The polynomial and B-spline basis functions are all of cubic degree in this figure. The grey lines in the background are results of methods discussed in other sections.

5.5. Machine learning techniques

The final class of methods discussed in this chapter are regression techniques based on machine learning approaches, which is a very promising tool for amplitude interpolation [294–299]. These techniques leverage the power of neural networks, approximator functions that are structured as a nested sequence of operations dependent on learnable parameters. Neural networks can be optimized to approximate any function of a set of inputs to arbitrary precision—given sufficient time and data. This is accomplished by stochastically minimizing the *loss function*, which characterizes how close the neural network results are to the desired ones. The optimization of neural network parameters is performed by evaluating the loss function on data subsets or batches and performing gradient descent to minimize the objective in an iterative way. In the present case, neural network trainings are framed as regression tasks, where the output of the network is trained to match a target given an input.

The main advantage of this approach for the interpolation problem is its versatility. Neural networks by default introduce a minimal bias in the interpolants they represent, so they can easily adapt to a wide variety of target distributions. Additionally, networks are not limited to a single training or dataset for their optimization. This enables their usage for adaptive interpolation, since after an initial training their weakest predictions can be used as a guideline for optimization through subsequent retrainings.

However, the prediction quality of neural amplitude surrogates declines substantially as particle multiplicity grows [296]. Their performance also decreases with the inclusion of beyond leading-order contributions; exploiting the known structure of infrared divergent QCD radiation can mitigate issues related to peaked integrands [297]. In this case, we tackle the increased amplitude complexity by including prior knowledge about physical symmetries into the learning task. We focus on the Lorentz symmetry and

work with architectures that are aware of the Lorentz invariance of the amplitudes. This feature is also present in all other interpolation methods discussed in this chapter, but there are several ways to incorporate it into neural networks. We explore two ways of enforcing Lorentz invariance. On the one hand, we train a *multilayer perceptron* (MLP) [300, 301] by feeding it only Lorentz invariant inputs. On the other hand, we use the *Lorentz-Equivariant Geometric Algebra Transformer* (L-GATr) [298, 299], a neural network whose operations are constrained to be equivariant (or covariant) with respect to Lorentz transformations.

To approximate the test functions, we train the networks on a_n as functions of the phase-space points using the test function from Eq. (5.9) as a loss function. We generate two sets of training data points, one sampled uniformly over the phase space, using a low-discrepancy Sobol sequence, and another using the leading-order-unweighted sampling described in Eq. (5.9). Training on the second set corresponds to option (1) from Sec. 5.1.4. We also generate unweighted samples for the testing data points. Additionally, we also build extra datasets from unweighted samples for *validation*, a routine checkup that is performed regularly during training to prevent overfitting and select the best performing model constructed during training. The validation set size is always 10% of that of the training dataset until it reaches a size of 4000 points; after that it remains constant for larger training sets.

For f_1 and f_3 our networks output a directly, while for f_2 and f_4 they output a/a_{LO} . Due to a larger range in the amplitude distribution, for f_5 we preprocess the amplitude targets by using logarithmic *standardization*:

$$\hat{a}_5 = \frac{\log(a_5) - \overline{\log(a_5)}}{\sigma_{\log(a_5)}}, \quad (5.40)$$

where $\overline{\log(y_i)}$ and $\sigma_{\log(y_i)}$ are the mean and standard deviation of the amplitude logarithm distributions over the whole dataset. In this case, the preprocessed amplitude \hat{a}_5 is taken as the argument for the loss in Eq. (5.9) instead of a_5 .

As for the inputs, all networks are trained on functions of the four-momenta of the sampled points, as it is observed that training directly on the original parametrization, e.g. as in Eq. (5.4), yields subpar results (see Fig. 5.18). Plus, L-GATr can only be trained on four-momenta inputs due to architecture constraints. We derive the four-momenta for each point from the sampled phase-space parameters. In doing so, we prepare the $t\bar{t}$ inputs for f_1 – f_4 so that the angle φ_t lies only in the range $[0, \pi]$. This decision allows all neural networks to skip learning about the parity symmetry of these functions during training and reduces the interpolation domain (see option (iii) from Sec. 5.1.4)).

All networks are trained for 5×10^5 steps with a batch size of 256, the Adam optimizer [302], and a Cosine Annealing scheduler [303]. Higher amounts of training iterations were observed to produce better performance (see Table 5.3). However, we choose to refrain from increasing the training time due to prohibitive training times on L-GATr. Due to instabilities during training, we refrain from applying early stopping and we perform validation checks every 300 iterations.

5.5.1. MLP

The MLP is built as a simple fully connected neural network with GELU activation functions [304]. We investigate two versions of it: $\text{MLP}(x)$, which takes \vec{x} as its inputs (the same as for the interpolation methods in the previous sections), and $\text{MLP}(s)$, which takes the Lorentz invariants built from pairs of momenta,

$$s_{ij} \equiv (p_i + p_j)^2, \quad \text{with } i \neq j, \quad (5.41)$$

amounting to 10 inputs for f_1 – f_4 and 6 inputs for f_5 , as opposed to $\text{MLP}(x)$, which takes 5 and 2 correspondingly.

Both MLP architectures consist of 5 hidden layers with 512 hidden channels each for all test functions, amounting to 10^6 learnable parameters. The inputs are preprocessed by taking their logarithms and performing standardization similar to the amplitude preprocessing in Eq. (5.40). We fix the maximum learning rate at 2×10^{-4} for f_1 – f_4 , and at 10^{-3} for f_5 . An average training on 10^6 points lasts around 40 minutes on an Nvidia A40 GPU. A scan over training runs with 10^6 data points is performed to select the hyperparameters, which comprise both the network structure and the learning parameters. This scan is executed by selecting an initial set of hyperparameter values and then increasing and decreasing each one by factors of 2 and 10 iteratively until the best performance is reached. We have observed that moderate changes in the network shape on a fixed parameter budget have a very small effect on network performance for all test functions. A summary of all the tests performed for hyperparameter optimization is shown in Table 5.3.

5.5.2. L-GATr

Equivariant neural networks constitute a very attractive option for any problem where symmetries are well defined [298, 299, 305–307]. These networks respect the spacetime symmetry properties of the data in every operation they perform. They do so by imposing the equivariance condition, defined as

$$f(\Lambda(x)) = \Lambda(f(x)), \quad (5.42)$$

where x is a network input, f is a network operation and Λ is a Lorentz transformation. By restricting the action of the network to equivariant maps, it does not need to learn the symmetry properties of the data during training and its range of operations gets reduced to only those allowed by the symmetry. This makes equivariant networks very efficient to train and capable of reaching high performance with low amounts of training data.

L-GATr is a neural network architecture that achieves equivariance by working in the spacetime geometric algebra representation [308]. A geometric algebra is generally defined as an extension of a vector space with an extra composition law: the geometric product. Given two vectors x and y , their geometric product can be expressed as the sum of a symmetric and an antisymmetric term

$$xy = \frac{\{x, y\}}{2} + \frac{[x, y]}{2}, \quad (5.43)$$

Network layers	Hidden channels	Training iterations	Learning rate	ε
3	512	5×10^5	10^{-3}	4.93×10^{-5}
3	512	5×10^5	2×10^{-4}	4.81×10^{-5}
3	512	5×10^5	10^{-5}	2.66×10^{-4}
3	512	5×10^6	10^{-3}	2.44×10^{-5}
3	512	5×10^6	2×10^{-4}	1.50×10^{-5}
3	512	5×10^6	10^{-5}	4.16×10^{-5}
5	512	5×10^5	10^{-3}	5.90×10^{-5}
5	512	5×10^5	2×10^{-4}	4.37×10^{-5}
5	512	5×10^5	10^{-5}	2.60×10^{-4}
5	512	5×10^6	10^{-3}	4.07×10^{-5}
5	512	5×10^6	2×10^{-4}	1.51×10^{-5}
5	512	5×10^6	10^{-5}	2.47×10^{-5}
10	1024	5×10^5	10^{-3}	9.72×10^{-5}
10	1024	5×10^5	2×10^{-4}	5.86×10^{-5}
10	1024	5×10^5	10^{-5}	2.25×10^{-4}

Table 5.3.: Summary of the MLP hyperparameter scan based on trainings with 10^6 data points for f_1 . The highlighted setup is the one that is used for the test function studies.

where the first term can be identified as the usual vector inner product and the second term represents a new operation called the *outer product*. The outer product allows for the combination of vectors to build higher-order geometric objects. In this particular case, $[x, y]$ is a bivector, which represents an element of the plane defined by the directions of x and y .

The spacetime algebra $\mathbb{G}^{1,3}$ can be built by introducing the geometric product on the Minkowski vector space $\mathbb{R}^{1,3}$. The geometric product in this space is fully specified by demanding that the basis elements of the vector space γ^μ satisfy the following anti-commutation relation:

$$\{\gamma^\mu, \gamma^\nu\} = 2g^{\mu\nu} . \quad (5.44)$$

This inner product establishes that the vectors γ^μ in the context of the spacetime algebra have the same properties as the gamma matrices from the Dirac algebra. Both algebras are tightly connected, with the Dirac algebra representing a complexification of the spacetime algebra.

With this notion in mind, we can now cover all unique objects in the spacetime algebra by taking antisymmetric products of the gamma matrices. A generic object of the algebra is called a multivector, and it can be expressed as

$$x = x^S 1 + x_\mu^V \gamma^\mu + x_{\mu\nu}^B \sigma^{\mu\nu} + x_\mu^A \gamma^\mu \gamma^5 + x^P \gamma^5 \quad \text{with} \quad \begin{pmatrix} x^S \\ x_\mu^V \\ x_{\mu\nu}^B \\ x_\mu^A \\ x^P \end{pmatrix} \in \mathbb{R}^{16}. \quad (5.45)$$

In this expression, the components of the multivector are divided in grades, defined by the number of gamma matrix indices that are needed to express them. Namely, $x^S 1$ constitutes the scalar grade, $x_\mu^V \gamma^\mu$ the vector grade, $x_{\mu\nu}^B \sigma^{\mu\nu}$ the geometric bilinear grade, $x_\mu^A \gamma^\mu \gamma^5$ the axial vector grade, and $x^P \gamma^5$ the pseudoscalar grade.

Apart from its extended representation power, the spacetime algebra offers a clear way to define equivariant transformations with respect to the Lorentz group on a wide range of geometric objects in Minkowski space. The main consideration is that Lorentz transformations on algebra elements act separately on each grade [298, 309]. As a consequence, any equivariant map must transform all components of a single grade in the same manner and allow for different grades to transform independently.

This guideline allows for an easy adaptation of standard neural network layers to equivariant operations in the algebra. In the case of L-GATr, this adaptation is performed on a transformer backbone. *Transformer* [310] is a neural network architecture that is well suited to deal with datasets organised as sets of particles, since its attention mechanism can be leveraged to capture correlations between them in an accurate way. L-GATr is built with equivariant versions of linear, attention, normalisation, and activation layers [310, 311]. It also includes a new layer MLPBlock featuring the geometric product to further increase the expressivity of the network. Its layer structure is

$$\begin{aligned} \bar{x} &= \text{LayerNorm}(x) \\ \text{AttentionBlock}(x) &= \text{Linear} \circ \text{Attention}(\text{Linear}(\bar{x}), \text{Linear}(\bar{x}), \text{Linear}(\bar{x})) + x \\ \text{MLPBlock}(x) &= \text{Linear} \circ \text{Activation} \circ \text{Linear} \circ \text{GP}(\text{Linear}(\bar{x}), \text{Linear}(\bar{x})) + x \\ \text{Block}(x) &= \text{MLPBlock} \circ \text{AttentionBlock}(x) \\ \text{L-GATr}(x) &= \text{Linear} \circ \text{Block} \circ \text{Block} \circ \dots \circ \text{Block} \circ \text{Linear}(x). \end{aligned} \quad (5.46)$$

All of these layers are redefined to operate on multi-vectors and restricted to act on algebra grades independently to ensure equivariance. Further details for each of the layers are provided in Refs. [298, 299].

Through this procedure, we build knowledge about the Lorentz symmetry into the architecture, but it can also be used to enforce awareness of the discrete symmetries described in Sec. 5.1.4 besides the parity invariance present in f_1 – f_4 (i.e., option (i) from Sec. 5.1.4). Being a transformer, L-GATr handles individual particle inputs independently and can enforce any exchange symmetry on individual particles. This is performed in practice by including common scalar labels to every set of particles that leave the amplitude invariant under permutation.

To operate with L-GATr, we need to embed inputs into the geometric algebra representation and undo said embedding once we obtain the outputs. For the interpolation problem, inputs always consist of particle 4-momenta p , which can be embedded into multi-vectors as

$$x_\mu^V = p_\mu \quad \text{and} \quad x^S = x_{\mu\nu}^T = x_\mu^A = x^P = 0. \quad (5.47)$$

Each particle is embedded to a different multivector with 16 components, resulting in the inputs being organized as sets of particles or tokens, 5 for f_1 – f_4 and 4 for f_5 . Each token also incorporates a distinct integer label as part of the inputs. The value for these labels is selected according to the permutation invariance pattern of each process we study. Specifically, we use the same token label for the particles in the initial state for all processes except f_2 , and we also use a shared label for the t and \bar{t} tokens for f_1 , f_3 and f_4 . This ensures that L-GATr also respects any instance of particle exchange symmetry. The token labels are not part of the multi-vectors, they are fed to the network as separate scalar inputs. Multi-vectors and scalars traverse the network through parallel tracks, mixing with each other only at the linear layers.

The output of the network is constructed through the use of a global token, which is introduced as an extra empty particle entry. This extra token holds no meaning at the input level, but at the output level its scalar component represents the estimator for the target amplitude. As for the input preprocessing, in L-GATr we divide them by the standard deviation over each of the particle momenta to prevent violating equivariance.

Our L-GATr build consists of a network with 8 attention blocks, 64 multivector channels, 32 scalar channels and 8 attention heads for all test functions, resulting in 7×10^6 learnable parameters. We set the maximum learning rate to 2×10^{-4} for f_1 – f_4 , and 5×10^{-4} for f_5 . An average L-GATr training on 10^6 points lasts around 40 hours on an Nvidia A40 GPU. As with the MLP, these values are chosen after a scan on trainings with 10^6 points to maximize performance on that data regime. We observe only a small effect in performance from changing the network shape on a fixed parameter budget across the different test functions. A summary of all the tests performed for hyperparameter optimization is shown in Table 5.4. The tests on this architecture are carried out using the same procedure as with the MLP, but they are more limited due to substantially longer training times.

5.5.3. Discussion

We show the results for the MLP and L-GATr networks in Fig. 5.18, where we display the results from training and testing on unweighted samples. MLP(x) performs consistently worse than MLP(s). Both MLP(s) and L-GATr display very similar performance for all test functions; MLP(s) is slightly better for f_5 , and f_2 in the low-precision limit, but slightly worse for f_4 in the high-precision limit.

We also observe a slowdown in improvement for all networks as we increase the amount of training data. This pattern has been empirically noted before [312–316]: the test loss goes down with the number of training points as a power law, but can eventually reach a lower bound, past which the performance cannot improve even with infinite data. The causes for this may involve insufficient training iterations, noise in the data, and overfitting. In our case, since our data is fully deterministic and we observe no

Blocks	Multivector channels	Scalar channels	Learning rate	ε
4	32	16	10^{-3}	7.70×10^{-5}
4	32	16	2×10^{-4}	1.07×10^{-4}
4	32	16	10^{-5}	3.31×10^{-4}
4	64	32	10^{-3}	8.79×10^{-5}
4	64	32	2×10^{-4}	6.44×10^{-5}
4	64	32	10^{-5}	1.72×10^{-4}
8	64	32	10^{-3}	6.99×10^{-5}
8	64	32	2×10^{-4}	5.15×10^{-5}
8	64	32	10^{-5}	6.57×10^{-5}

Table 5.4.: Summary of the L-GATr hyperparameter scan based on trainings with 10^6 data points for f_1 . All network builds are trained for 5×10^5 iterations. The highlighted setup is the one that is used for the test function studies.

overfitting in the large dataset regime, we theorize that the main bottleneck is training time. Indeed, the left panel of Fig. 5.16³⁵ and Table 5.3 show that longer trainings on the MLP substantially improves the performance in the large data limit. Despite these observations, a dedicated study to better understand this limitation in the context of amplitude interpolation may still be needed.

Comparing with other methods, machine learning algorithms surpass other interpolation methods in the low-to-medium precision regime for f_1 – f_4 , and they are only overtaken in the high precision limit, after machine learning results stop improving. The advantage over other approaches is the greatest for f_2 and f_4 , which represent the more complicated higher order corrections to amplitudes. This signals potential utility of machine learning approaches for the interpolation of more complex multi-loop amplitudes.

The only front where neural networks underperform is in the case of f_5 . We note that this is the only 2-dimensional function tested, and that seeing less benefits from neural networks in lower dimensions should not be surprising.

Another important result that we obtain from this study is that neural network training appears to be insensitive to the distribution of points it is trained on. As can be seen from the right panel of Fig. 5.16, we can train the MLP on either uniform or unweighted samples and get similar performance for all training set sizes. This means that we can e.g. train our algorithms on naively sampled datasets, while targeting evaluation on physically motivated distributions without a significant performance degradation.

³⁵In this plot the degraded performance at the 10^3 – 10^4 data point range is due to overfitting, which is more severe in this long training regime due to the slower learning rate reduction set up by the scheduler.

From Fig. 5.17 we observe that dividing by the leading-order amplitudes a_{LO} brings a small but stable improvement for f_2 and f_4 , unlike for polynomials and sparse grids, where the advantage disappears with more data.

Finally, we note that it should be possible to adaptively add to the training dataset focusing on regions that are poorly estimated in the initial stages of the training. This might work especially well combined with methods that provide uncertainty estimates on predictions, such as Bayesian neural networks [296, 317–320] or repulsive ensembles [321, 322].

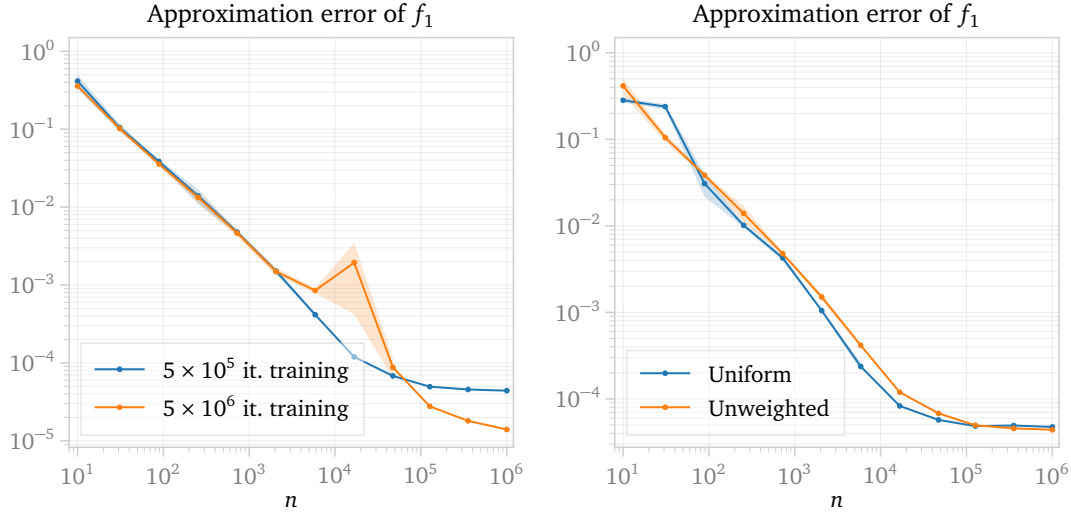


Figure 5.16.: Comparison of the effect of training duration (left) and sampling method choice (right) on the f_1 MLP performance. Error bands are obtained as the standard deviation of the test function estimations from 3 independent runs.

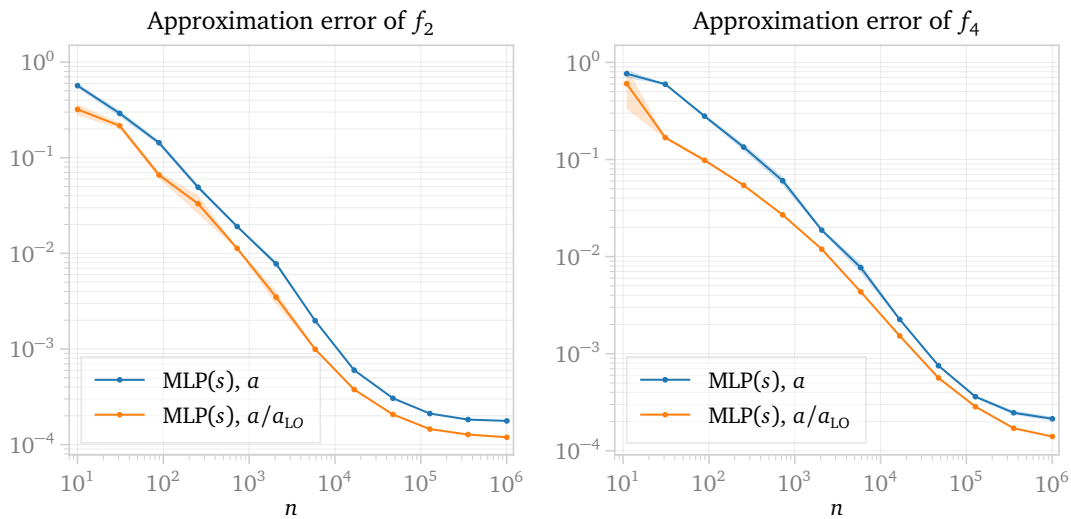


Figure 5.17.: Comparison of f_2 (left) and f_4 (right) MLP trainings on the a/a_{LO} ratio versus the loop amplitudes a . Error bands are obtained as the standard deviation of the test function estimations from 3 independent runs.

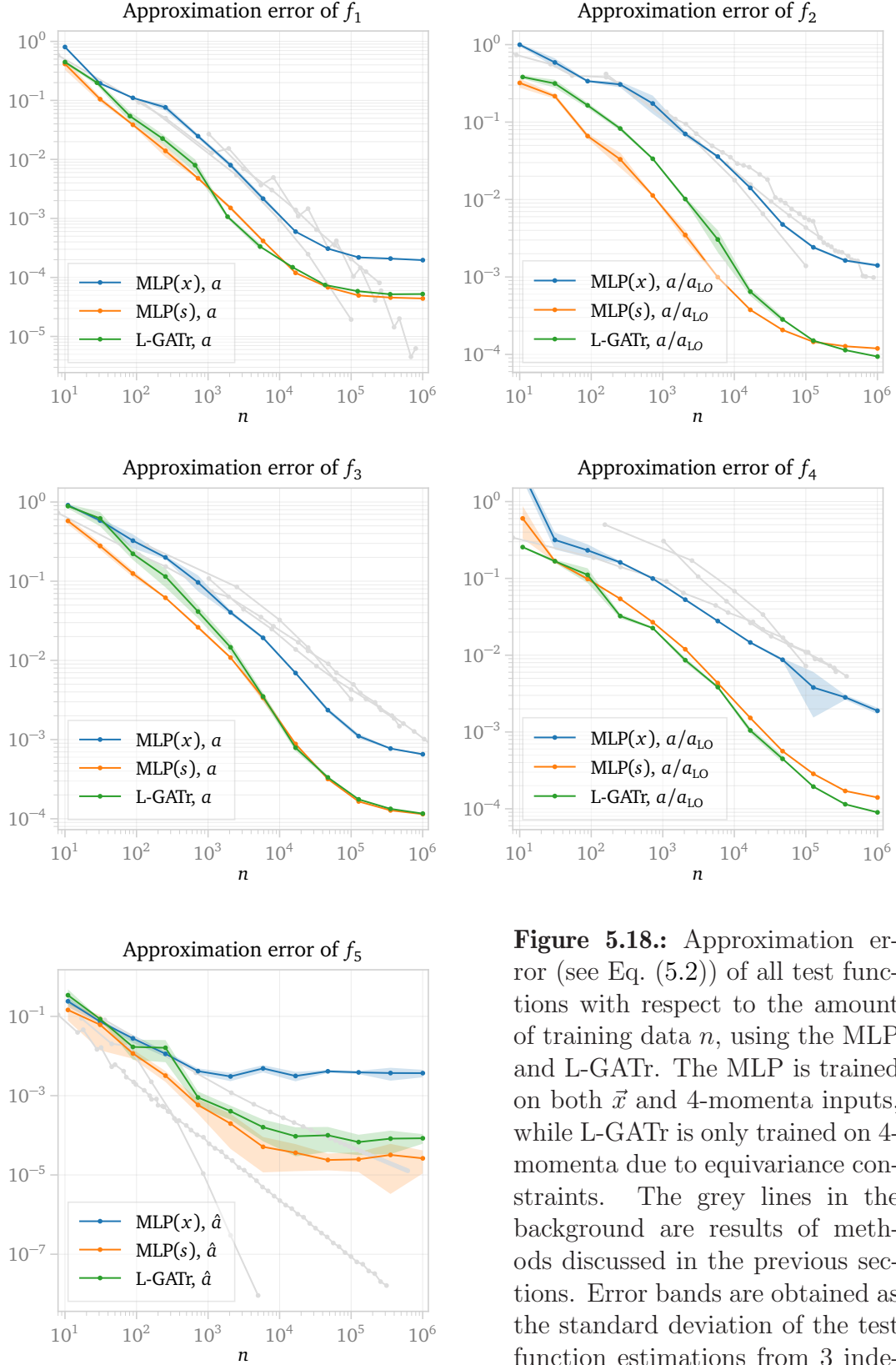


Figure 5.18.: Approximation error (see Eq. (5.2)) of all test functions with respect to the amount of training data n , using the MLP and L-GATr. The MLP is trained on both \vec{x} and 4-momenta inputs, while L-GATr is only trained on 4-momenta due to equivariance constraints. The grey lines in the background are results of methods discussed in the previous sections. Error bands are obtained as the standard deviation of the test function estimations from 3 independent runs.

5.6. Conclusions

In this chapter we have presented a detailed investigation of several frameworks to interpolate multidimensional functions, focusing on scattering amplitudes depending on a number of phase-space variables. We have used amplitudes related to $t\bar{t}H$ production (5-dimensional phase space, test functions f_1 – f_4) and Higgs+jet production (2-dimensional phase space, test function f_5) at the LHC as test functions, and applied various ways of polynomial interpolation, B-spline interpolation, spatially adaptive sparse grids, and regression based on machine learning techniques, using a standard multi-layer perceptron and the Lorentz-Equivariant Geometric Algebra Transformer. Considering that the evaluation of loop amplitudes is costly, the main performance measure is how the approximation error of the different interpolation methods scales with the number of data points.

To measure the approximation error we notice that loop amplitudes typically have peaks or show a steep rise towards the phase-space boundaries. However, the contribution of such regions to physical observables might be small because it is suppressed by the phase-space density or by parton distributions. With this in mind, we have used physically motivated error metrics that weigh the approximation errors in certain phase-space regions proportionally to their contribution to the total cross section.

For the 5-dimensional test functions, we find that machine learning techniques significantly outperform the classical interpolation methods at 10^{-2} to 10^{-4} approximation error, in some cases by several orders of magnitude. For the 2-dimensional case, we see that most methods perform well enough, but if very high precision is required, sparse grids are the best choice. In fact, for all test functions we see that the performance of machine learning approaches starts to stagnate after a certain precision threshold, while classical interpolation approaches scale smoothly. Since this threshold is fairly high, this still leaves machine learning techniques as the best practical choice in high dimensions, but does call for further investigation.

For test functions representing a higher order correction to amplitudes, we see an improvement from training on the ratio a/a_{LO} instead of the raw loop amplitude a . For some methods the improvement is only significant in the low-data regime, while the neural network approaches benefit from it at all training sizes.

Another important measure for the usefulness of a particular method is its extendibility. Since it is difficult to predict how much data is needed to reach the desired error target, it is likely that more training data needs to be added between validations. Both the adaptive methods and neural networks provide this option. Spatially adaptive sparse grids are particularly flexible: they allow for the addition of single points at a time and for an assessment of which regions are approximated the worst, and thus need more points. Neural networks are fairly insensitive to the data-point distribution, which allows for more data to be easily added, at the cost of a repeated training.

In summary, the investigations show that the construction of multidimensional amplitude interpolation frameworks with a sufficient precision for current and upcoming collider experiments is feasible, with machine learning being the most promising tool for it. Therefore, numerical calculations for multi-scale loop amplitudes continue to have a promising future. Finally, implementations of the described methods, as well as the test functions used for the benchmarks, can be found at github.com/OlssonA/interpolating_amplitudes.

CHAPTER 6

Final Conclusion and Outlook

In this thesis we have presented techniques for the computation of multi-loop amplitudes and Feynman integrals, with a focus on processes that involve many kinematic scales. The developed methods mainly concern strategies for numerical evaluation, that are important for dealing with increasingly complicated integrals appearing at higher perturbative orders in processes with multiple mass scales, or with many particles in the final state. The main results are presented at the end of Chapter 3, where the PYSECDEC and FTINT programs are described, in Chapter 4, on the topic of positive integrands for singularity resolution of Feynman integrals, and in Chapter 5, which concerns the construction of interpolation grids for multidimensional amplitudes.

The latest developments of PYSECDEC, and the new program FTINT, is presented in Chapter 3. The new versions of PYSECDEC focus on amplitude computations through adaptive evaluations of weighted sums. This feature is also useful for computing individual resolved integrals from Chapter 4, as they are always expressed as a sum of two or more integrals. In FTINT, the usage of PYSECDEC enables the numerical evaluation of massive flow-time integrals up to the three-loop level. As other aspects of multi-loop computations in flowed QFT, such as diagram generation and IBP reduction, are already available in a mature state of automation, FTINT provides the missing piece for further phenomenological applications of the perturbative approach to the gradient-flow formalism.

In Chapter 4, a new procedure for decomposing Feynman integrals into a sum of integrals over positive integrands multiplied by complex prefactors is described. This removes the need for contour deformation for integration in the physical region. For integrals with many internal lines, which are present in amplitudes with many external particles and at higher loops, this has accelerated the integration time by up to four orders of magnitude. For integrals with few internal lines the improvements are more modest, but still significant for extreme kinematic configurations, for example when internal masses are very small or in the high-energy region. Additionally, in particular for massive integrals, the highest attainable precision is shown to be improved for

the resolved integrals. The resolution procedure is shown to work for both massless and massive integrals up to three loops. The massive examples include integrals with complicated underlying geometries, with the equal-mass banana integral being related to a Calabi-Yau 2-fold. It is shown how such properties pose no inherent limitations on the resolution procedure, which is promising for phenomenological applications in the future. A limitation in the massive case, is that geometric visualisation has been used to understand how to dissect the integration region, implying a limitation on the number of propagators. Developing a more general strategy for the massive case, and automating the procedure in a code that can be interfaced to programs such as PYSECDEC, is on-going work.

In Chapter 5, a framework for the interpolation of virtual amplitudes is presented, which is a crucial part of the multi-loop workflow. A variety of state-of-the-art approximation methods have been investigated in the context of multi-scale amplitude interpolation. The machine-learning based approach significantly outperforms the classical interpolation methods in the multidimensional case. It has been shown that sufficient accuracy for phenomenological application is obtainable for five-dimensional $t\bar{t}H$ amplitudes at LO and NLO. We have also shown that adding noise to the training data does not significantly worsen the performance of the approximation, which is important for multi-loop applications where obtaining high accuracy is expensive. Additionally, it is explained in detail how a physical error estimate is obtained by propagating the error on the amplitude to observables. While the classical interpolation methods struggle with the five-dimensional test functions, they perform well in the lower-dimensional case. They also fill other niches. For example, B-splines are much faster to evaluate than other interpolants, and sparse grids are completely adaptive which is useful when it is hard to predict the required amount of training data.

In summary, the methods developed in this thesis demonstrate the feasibility of numerical approaches for the computation of multi-loop amplitudes that involve many kinematic scales. Such amplitudes will be increasingly important in the near future, for example at the high-luminosity phase of the LHC, where the increased amount of data will enable accurate measurements of processes with many particles in the final state. Additionally, at future lepton colliders, electroweak corrections that involve many mass scales will become more relevant. The next objective is to implement the singularity resolution procedure in PYSECDEC, such that it can be used together with the interpolation framework in a wide variety of phenomenological applications.

APPENDIX A

Test functions

In this appendix we provide some details on the test functions and the phase spaces of the associated amplitudes, used to asses the performance of the interpolation methods from Chapter 5.

A.1. Phase-space parametrisation for $pp \rightarrow t\bar{t}H$

The test functions $f_1 - f_4$ are based on $t\bar{t}H$ amplitudes up to NLO in the quark and gluon channels. They are defined on a $2 \rightarrow 3$ phase space where there are three angular variables and two variables of energy type:

$$\begin{aligned} d\Phi_{t\bar{t}H} = & \frac{1}{2^{10}\pi^4 \hat{s} s_{t\bar{t}}} \sqrt{\lambda(s_{t\bar{t}}, m_t^2, m_t^2)} \sqrt{\lambda(\hat{s}, s_{t\bar{t}}, m_H^2)} \times \\ & \Theta(\sqrt{\hat{s}} - 2m_t - m_H) \Theta(s_{t\bar{t}} - 4m_t^2) \Theta([\sqrt{\hat{s}} - m_H]^2 - s_{t\bar{t}}) ds_{t\bar{t}} d\Omega_{t\bar{t}} \sin\theta_H d\theta_H, \end{aligned} \quad (\text{A.1})$$

with θ_H being the polar angle of the Higgs boson relative to the beam axis, $d\Omega_{t\bar{t}} = \sin\theta_t d\theta_t d\varphi_t$ and λ being the Källén function, $\lambda(a, b, c) = a^2 + b^2 + c^2 - 2ab - 2bc - 2ca$, depending on the variables

$$\hat{s} = (p_3 + p_4 + p_5)^2, \quad s_{t\bar{t}} = (p_3 + p_4)^2 \quad (\text{A.2})$$

and the masses. As the production threshold of the $t\bar{t}H$ system is located at $s_0 = (2m_t + m_H)^2$, a convenient variable for a scan in partonic energy is

$$\beta^2 = 1 - \frac{s_0}{\hat{s}}, \quad (\text{A.3})$$

such that $\beta^2 = 0$ at the production threshold and $\beta^2 \rightarrow 1$ in the high energy limit. For more details we refer to Ref. [16].

A.2. Amplitude definitions

f_1 : **leading-order** $q\bar{q} \rightarrow t\bar{t}H$

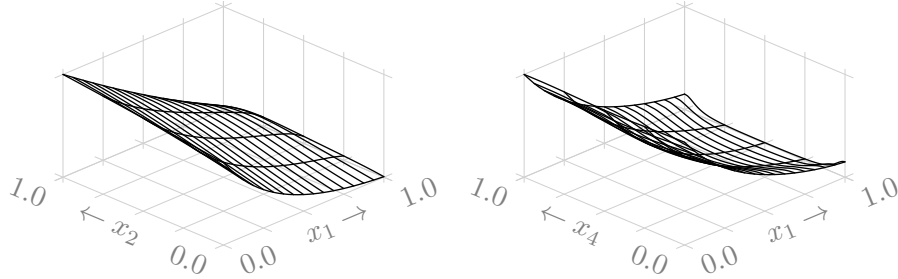
The squared amplitude and the phase-space parameters are already defined but we detail some discrete amplitude symmetries. The most interesting ones are the symmetries under $\varphi_t \rightarrow -\varphi_t$ (parity invariance) and the swap $\{q, t\} \leftrightarrow \{\bar{q}, \bar{t}\}$. These two symmetries are present in all higher order corrections to this amplitude too; the leading order is additionally symmetric under the swap $q \leftrightarrow \bar{q}$. In terms of the variables x_i , this works out to

$$\varphi_t \rightarrow -\varphi_t : \quad a_1(x_5) = a_1(1 - x_5), \quad (\text{A.4})$$

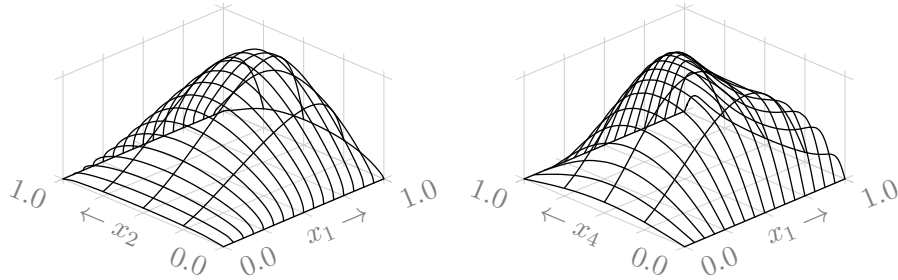
$$\{q, t\} \leftrightarrow \{\bar{q}, \bar{t}\} : \quad a_1(x_3, x_4) = a_1(1 - x_3, 1 - x_4), \quad (\text{A.5})$$

$$q \leftrightarrow \bar{q} : \quad a_1(x_3, x_5) = a_1(1 - x_3, x_5 + 1/2). \quad (\text{A.6})$$

As an illustration, slices of the amplitude a_1 in x_1 - x_2 and x_1 - x_4 space are as follows:



The same slices for the function f_1 are:



f_2 : **next-to-leading-order** $q\bar{q} \rightarrow t\bar{t}H$

This function is a one loop contribution to the squared $q\bar{q} \rightarrow t\bar{t}H$ amplitude. The test function has the Coulomb singularity subtracted off, and is defined by

$$a_2 = 2 \operatorname{Re} \left[\langle \mathcal{M}_0^{q\bar{q}t\bar{t}H} | \mathcal{M}_1^{q\bar{q}t\bar{t}H} \rangle + \frac{\pi^2}{\beta_{t\bar{t}}} \langle \mathcal{M}_0^{q\bar{q}t\bar{t}H} | \mathbf{T}_{t\bar{t}} | \mathcal{M}_0^{q\bar{q}t\bar{t}H} \rangle \right], \quad (\text{A.7})$$

$$f_2 = a_2 \times \left| \frac{d\Phi_{t\bar{t}H}}{d(\operatorname{frac}_{s_{t\bar{t}}}, \theta_H, \theta_t, \varphi_t)} \right| \times \frac{1}{2\hat{s}} \frac{d\rho_{q\bar{q}}}{d\beta^2} \times J_{t\bar{t}H}, \quad (\text{A.8})$$

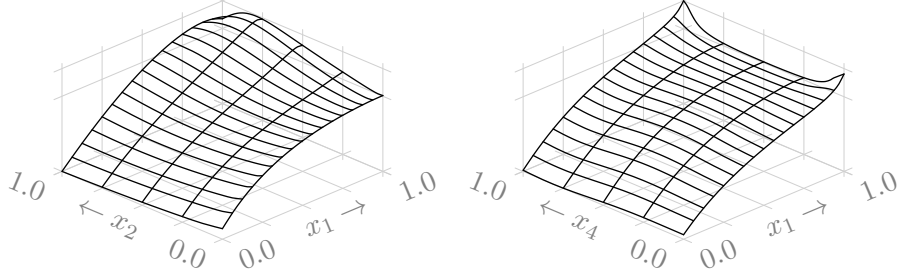
where $\mathbf{T}_{t\bar{t}}$ is a colour operator defined in Ref. [323] and $\beta_{t\bar{t}}$ is the velocity of the $t\bar{t}$ system, in our variables given by

$$\beta_{t\bar{t}}^2 \equiv 1 - \left[1 + \operatorname{frac}_{s_{t\bar{t}}} \left(\left(\frac{1 + \frac{1}{2} \frac{m_H}{m_t}}{\sqrt{1 - \beta^2}} - \frac{1}{2} \frac{m_H}{m_t} \right)^2 - 1 \right) \right]^{-1}. \quad (\text{A.9})$$

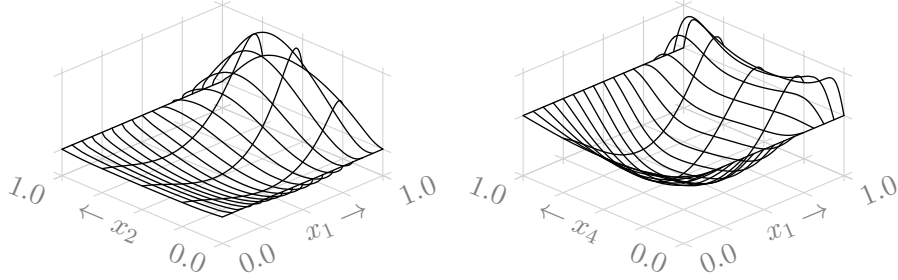
The phase-space parameters are the same as for f_1 , but we only have two of the symmetries in this case:

$$a_2(x_5) = a_2(1 - x_5), \quad \text{and} \quad a_2(x_3, x_4) = a_2(1 - x_3, 1 - x_4). \quad (\text{A.10})$$

Slices of the amplitude a_2 in x_1 - x_2 and x_1 - x_4 space are as follows:



The same slices for the function f_2 are:

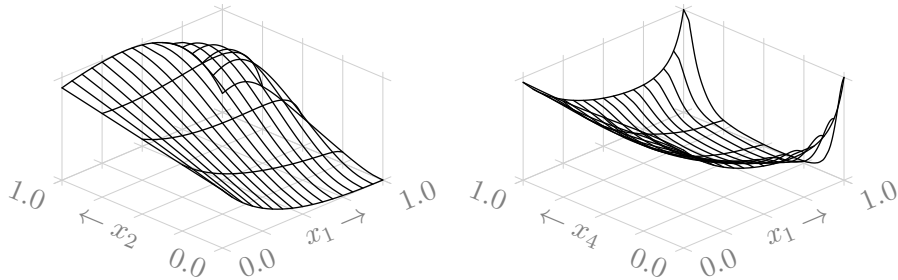


f_3 : leading-order $gg \rightarrow t\bar{t}H$

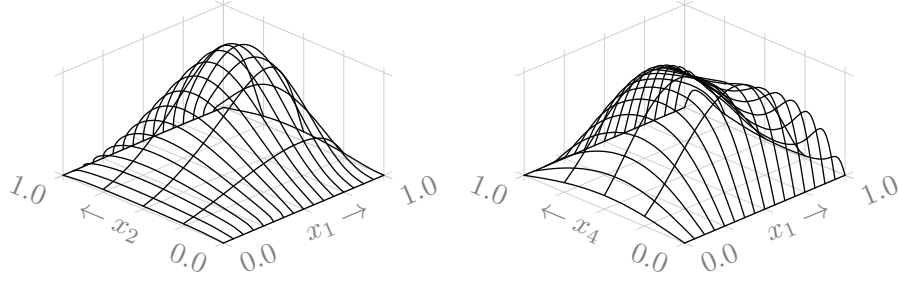
This is the leading-order contribution to the squared $gg \rightarrow t\bar{t}H$ amplitude. The test function is defined by

$$a_3 = \langle \mathcal{M}_0^{gg\bar{t}tH} | \mathcal{M}_0^{gg\bar{t}tH} \rangle, \quad f_3 = a_3 \times \left| \frac{d\Phi_{t\bar{t}H}}{d(\text{frac}_{s_{t\bar{t}}}, \theta_H, \theta_t, \varphi_t)} \right| \times \frac{1}{2\hat{s}} \frac{d\rho_{gg}}{d\beta^2} \times J_{t\bar{t}H}, \quad (\text{A.11})$$

The amplitude a_3 possesses the same symmetries as a_1 , given in Eq. (A.6). Slices of the amplitude a_3 in x_1 - x_2 and x_1 - x_4 space are as follows:



The same slices for the function f_3 are:



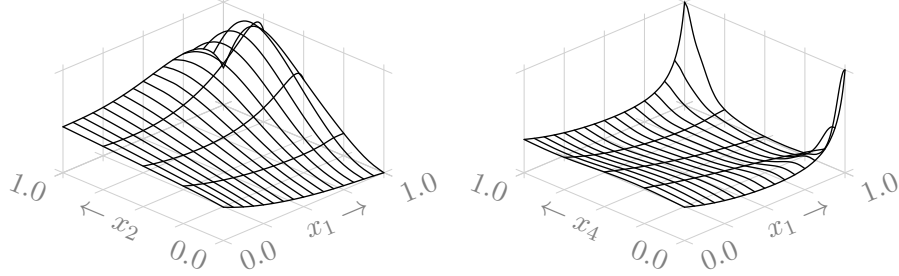
f_4 : next-to-leading-order $gg \rightarrow t\bar{t}H$

This function is a one loop contribution to the squared $gg \rightarrow t\bar{t}H$ amplitude. In the same way as for a_2 , the test function has the Coulomb singularity subtracted off, and is defined by

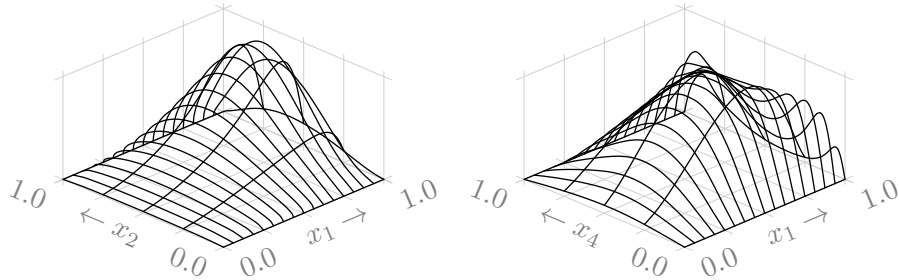
$$a_4 = 2\text{Re} \left[\langle \mathcal{M}_0^{gg t\bar{t}H} | \mathcal{M}_1^{gg t\bar{t}H} \rangle + \frac{\pi^2}{\beta_{t\bar{t}}} \langle \mathcal{M}_0^{gg t\bar{t}H} | \mathbf{T}_{t\bar{t}} | \mathcal{M}_0^{gg t\bar{t}H} \rangle \right], \quad (\text{A.12})$$

$$f_4 = a_4 \times \left| \frac{d\Phi_{t\bar{t}H}}{d(\text{frac}_{s_{t\bar{t}}, \theta_H, \theta_t, \varphi_t})} \right| \times \frac{1}{2\hat{s}} \frac{d\rho_{gg}}{d\beta^2} \times J_{t\bar{t}H}, \quad (\text{A.13})$$

where $\beta_{t\bar{t}}$ is given in Eq. (A.9), and the rest is the same as for f_3 . The amplitude a_4 possesses the same symmetries as a_3 and a_1 , given in Eq. (A.6). Slices of the amplitude a_4 in x_1 - x_2 and x_1 - x_4 space are as follows:



The same slices for the function f_4 are:



f_5 : leading-order $gg \rightarrow Hg$

This is the leading-order contribution to the squared $gg \rightarrow Hg$ amplitude. The test function depends on two phase-space variables and is defined by

$$a_5 = \langle \mathcal{M}_1^{ggHg} | \mathcal{M}_1^{ggHg} \rangle, \quad f_5 = a_5 \times \frac{d\Phi_{Hg}}{d\theta_H} \times \frac{1}{2\hat{s}} \frac{d\rho_{gg}}{d\beta^2} \times J_{Hg}, \quad (\text{A.14})$$

with the phase-space parameters set as

$$\beta^2 = \frac{33}{100} + \frac{66}{100}x_1, \quad \theta_H = \theta_0 + (\pi - 2\theta_0)x_2, \quad (\text{A.15})$$

$$J_{Hg} = \left| \frac{d(\beta^2, \theta_H)}{d\vec{x}} \right| = \frac{66}{100}(\pi - 2\theta_0), \quad (\text{A.16})$$

and the phase-space density being

$$\frac{d\Phi_{Hg}}{d\theta_H} = \frac{1}{16\pi} \frac{1}{\hat{s}} \sqrt{\lambda(\hat{s}, m_H^2, 0)} \sin \theta_H = \frac{\beta^2 \sin \theta_H}{16\pi}. \quad (\text{A.17})$$

The introduction of θ_0 as a cutoff is needed because a_5 diverges as $1/\sin^2 \theta_H$ at $\theta_H \rightarrow 0$ and $\theta_H \rightarrow \pi$. We choose not to interpolate the region around the divergence, because in practical calculations it should be regulated by infrared subtraction or appropriate kinematic cuts; we only consider the phase-space region where the transverse momentum p_T of the Higgs boson H is greater than a cutoff $p_{T,0}$. Then:

$$\sin \theta_0 = p_{T,0} \frac{2\sqrt{\hat{s}}}{\sqrt{\lambda(\hat{s}, m_H^2, 0)}} = 2 \frac{p_{T,0}}{m_H} \frac{\sqrt{1 - \beta^2}}{\beta^2}. \quad (\text{A.18})$$

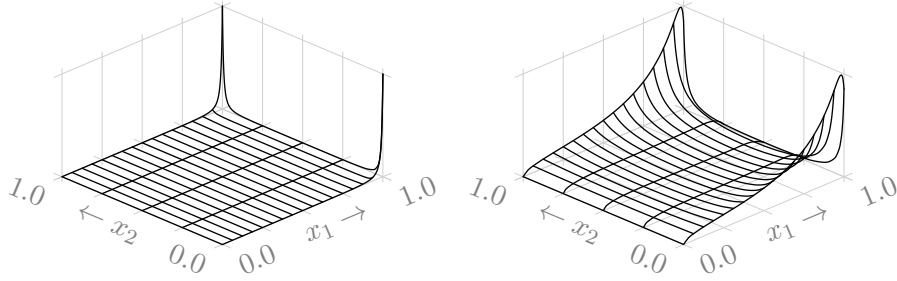
We choose $p_{T,0}$ corresponding to the lower boundary of the β^2 region from Eq. (A.15):

$$\frac{p_{T,0}}{m_H} = \frac{1}{2} \frac{\beta_{\min}^2}{\sqrt{1 - \beta_{\min}^2}}, \quad (\text{A.19})$$

such that at $\beta^2 < \beta_{\min}^2$ no phase-space point passes the p_T cut. This works out to $p_{T,0} \approx 25$ GeV. The amplitude a_5 is symmetric under the swap of the incoming gluons, i.e.,

$$a_5(x_2) = a_5(1 - x_2). \quad (\text{A.20})$$

The amplitude a_5 (left) and the function f_5 (right) depend on x_1 and x_2 as follows:



Acknowledgements

First of all, I would like to express my gratitude to my supervisor Gudrun Heinrich for granting me the opportunity to work on my doctorate at the ITP in Karlsruhe. During my time as a doctoral student, she has involved me in many exciting research projects and always made sure that I have interesting problems to work on. Throughout many discussions and meetings her wide expertise in the field has provided me with invaluable guidance.

I also want to thank all my other colleagues in Karlsruhe and elsewhere, without whom this work would not have been possible. I am especially grateful to Stephen Jones, for the opportunity to work together on the 'No CD' project, and also to Robert Harlander, for the collaboration on FTINT and for receiving me at RWTH Aachen during my stay there. A special thanks also to Vitaly Magerya, who for a big part of my PhD acted as a mentor and taught me many things about physics and programming, and also to Bakul Agarwal, for our continued collaboration and for patiently answering an endless amount of my questions.

To everyone who helped proofread and provided feedback on this thesis, I am very grateful. In particular, to Augustin Vestner for assisting me with translating the abstract into German, to Matthias Kerner and Bakul Agarwal for valuable comments on the chapter on Feynman integrals, to Sauro Carlotti for the feedback on the introduction, and of course again to Gudrun Heinrich, whose recurring feedback helped me stay on track during the writing process.

I also greatly appreciate the time I got to spend with all my friends and colleagues at the ITP. A special thanks goes to my current and former office mates; Lucas Kunz, Stefan Kiebacher, Sauro Carlotti and Jens Braun, for upholding a friendly and productive work environment. I am also very grateful to my friends outside KIT, who made my time in Karlsruhe truly enjoyable.

Finally, I want to thank my family who have, and continue to, support my doctoral career abroad.

References

- [1] **ATLAS** Collaboration, G. Aad *et al.*, “*Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC,*” **Phys. Lett. B** **716** (2012) 1–29.
- [2] **CMS** Collaboration, S. Chatrchyan *et al.*, “*Observation of a New Boson at a Mass of 125 GeV with the CMS Experiment at the LHC,*” **Phys. Lett. B** **716** (2012) 30–61.
- [3] **Particle Data Group** Collaboration, S. Navas *et al.*, “*Review of particle physics,*” **Phys. Rev. D** **110** no. 3, (2024) 030001.
- [4] **UA1** Collaboration, G. Arnison *et al.*, “*Experimental Observation of Isolated Large Transverse Energy Electrons with Associated Missing Energy at $\sqrt{s} = 540$ GeV,*” **Phys. Lett. B** **122** (1983) 103–116.
- [5] **UA1** Collaboration, G. Arnison *et al.*, “*Experimental Observation of Lepton Pairs of Invariant Mass Around 95-GeV/c**2 at the CERN SPS Collider,*” **Phys. Lett. B** **126** (1983) 398–410.
- [6] **D0** Collaboration, S. Abachi *et al.*, “*Observation of the top quark,*” **Phys. Rev. Lett.** **74** (1995) 2632–2637.
- [7] **CDF** Collaboration, F. Abe *et al.*, “*Observation of top quark production in $\bar{p}p$ collisions,*” **Phys. Rev. Lett.** **74** (1995) 2626–2631.
- [8] X. Fan, T. G. Myers, B. A. D. Sukra, and G. Gabrielse, “*Measurement of the Electron Magnetic Moment,*” **Phys. Rev. Lett.** **130** no. 7, (2023) 071801.
- [9] D. J. E. Marsh, “*Axion Cosmology,*” **Phys. Rept.** **643** (2016) 1–79.
- [10] G. Apollinari, O. Brüning, T. Nakamoto, and L. Rossi, “*High Luminosity Large Hadron Collider HL-LHC,*” **CERN Yellow Rep. no. 5**, (2015) 1–19.
- [11] **CMS** Collaboration, “*Highlights of the HL-LHC physics projections by ATLAS and CMS,*”.
- [12] G. Heinrich, S. P. Jones, M. Kerner, V. Magerya, A. Olsson, and J. Schlenk, “*Numerical scattering amplitudes with pySecDec,*” **Comput. Phys. Commun.** **295** (2024) 108956.
- [13] R. V. Harlander, T. Nellopoulos, A. Olsson, and M. Wesle, “*ftint: Calculating gradient-flow integrals with pySecDec,*” **Comput. Phys. Commun.** **306** (2025) 109384.

- [14] S. Jones, A. Olsson, and T. Stone, “*Positive Integrands from Feynman Integrals in the Minkowski Regime*,” 6, 2025. <https://arxiv.org/abs/2506.24073>.
- [15] V. Bresó, G. Heinrich, V. Magerya, and A. Olsson, “*Interpolating amplitudes*,” 12, 2024. <https://arxiv.org/abs/2412.09534>.
- [16] B. Agarwal, G. Heinrich, S. P. Jones, M. Kerner, S. Y. Klein, J. Lang, V. Magerya, and A. Olsson, “*Two-loop amplitudes for $t\bar{t}H$ production: the quark-initiated N_f -part*,” **JHEP** **05** (2024) 013. [Erratum: JHEP 06, 142 (2024)].
- [17] M. E. Peskin and D. V. Schroeder, *An Introduction to quantum field theory*. Addison-Wesley, Reading, USA, 1995.
- [18] L. D. Faddeev and V. N. Popov, “*Feynman Diagrams for the Yang-Mills Field*,” **Phys. Lett. B** **25** (1967) 29–30.
- [19] C. Becchi, A. Rouet, and R. Stora, “*Renormalization of the Abelian Higgs-Kibble Model*,” **Commun. Math. Phys.** **42** (1975) 127–162.
- [20] I. V. Tyutin, “*Gauge Invariance in Field Theory and Statistical Physics in Operator Formalism*,” 1975.
- [21] M. Schwartz, *Quantum Field Theory and the Standard Model*. Cambridge University Press, 2013.
- [22] P. W. Higgs, “*Broken Symmetries and the Masses of Gauge Bosons*,” **Phys. Rev. Lett.** **13** (Oct, 1964) 508–509.
- [23] F. Englert and R. Brout, “*Broken Symmetry and the Mass of Gauge Vector Mesons*,” **Phys. Rev. Lett.** **13** (1964) 321–323.
- [24] J. Campbell, J. Huston, and F. Krauss, *The Black Book of Quantum Chromodynamics: A Primer for the LHC Era*. Oxford University Press, 12, 2017.
- [25] G. Altarelli and G. Parisi, “*Asymptotic Freedom in Parton Language*,” **Nucl. Phys. B** **126** (1977) 298–318.
- [26] Y. L. Dokshitzer, “*Calculation of the Structure Functions for Deep Inelastic Scattering and e^+e^- Annihilation by Perturbation Theory in Quantum Chromodynamics*,” **Sov. Phys. JETP** **46** (1977) 641–653.
- [27] V. N. Gribov and L. N. Lipatov, “*Deep inelastic $e p$ scattering in perturbation theory*,” **Sov. J. Nucl. Phys.** **15** (1972) 438–450.
- [28] J. C. Collins, D. E. Soper, and G. F. Sterman, “*Transverse Momentum Distribution in Drell-Yan Pair and W and Z Boson Production*,” **Nucl. Phys. B** **250** (1985) 199–224.
- [29] J. C. Collins, D. E. Soper, and G. F. Sterman, “*Factorization for Short Distance Hadron - Hadron Scattering*,” **Nucl. Phys. B** **261** (1985) 104–142.
- [30] R. J. Eden, P. V. Landshoff, D. I. Olive, and J. C. Polkinghorne, *The Analytic S-Matrix*. Cambridge University Press, 1966.

- [31] M. Beneke and V. A. Smirnov, “*Asymptotic expansion of Feynman integrals near threshold*,” *Nucl. Phys. B* **522** (1998) 321–344.
- [32] R. Harlander, S. Y. Klein, and M. C. Schaaf, “*FeynGame-2.1 – Feynman diagrams made easy*,” *PoS EPS-HEP2023* (2024) 657.
- [33] L. Bündgen, R. V. Harlander, S. Y. Klein, and M. C. Schaaf, “*FeynGame 3.0*,” *Comput. Phys. Commun.* **314** (2025) 109662.
- [34] W. Beenakker, S. Dittmaier, M. Kramer, B. Plumper, M. Spira, and P. M. Zerwas, “*NLO QCD corrections to t anti- t H production in hadron collisions*,” *Nucl. Phys. B* **653** (2003) 151–203.
- [35] S. Dawson, C. Jackson, L. H. Orr, L. Reina, and D. Wackeroth, “*Associated Higgs production with top quarks at the large hadron collider: NLO QCD corrections*,” *Phys. Rev. D* **68** (2003) 034022.
- [36] J. C. Taylor, “*Ward Identities and Charge Renormalization of the Yang-Mills Field*,” *Nucl. Phys. B* **33** (1971) 436–444.
- [37] A. A. Slavnov, “*Ward Identities in Gauge Theories*,” *Theor. Math. Phys.* **10** (1972) 99–107.
- [38] A. Deur, “*The QCD Running Coupling*,”
<https://arxiv.org/abs/2502.06535>.
- [39] CMS Collaboration, A. Hayrapetyan *et al.*, “*Measurement of multijet azimuthal correlations and determination of the strong coupling in proton-proton collisions at $\sqrt{s} = 13$ TeV*,” *Eur. Phys. J. C* **84** no. 8, (2024) 842.
- [40] P. A. Baikov, K. G. Chetyrkin, and J. H. Kühn, “*Five-Loop Running of the QCD Coupling Constant*,” *Phys. Rev. Lett.* **118** no. 8, (2017) 082002.
- [41] T. Luthe, A. Maier, P. Marquard, and Y. Schröder, “*Towards the five-loop Beta function for a general gauge group*,” *JHEP* **07** (2016) 127.
- [42] F. Herzog, B. Ruijl, T. Ueda, J. A. M. Vermaseren, and A. Vogt, “*The five-loop beta function of Yang-Mills theory with fermions*,” *JHEP* **02** (2017) 090.
- [43] T. Luthe, A. Maier, P. Marquard, and Y. Schröder, “*The five-loop Beta function for a general gauge group and anomalous dimensions beyond Feynman gauge*,” *JHEP* **10** (2017) 166.
- [44] K. G. Chetyrkin, G. Falcioni, F. Herzog, and J. A. M. Vermaseren, “*Five-loop renormalisation of QCD in covariant gauges*,” *JHEP* **10** (2017) 179.
- [45] G. ’t Hooft and M. J. G. Veltman, “*Regularization and Renormalization of Gauge Fields*,” *Nucl. Phys. B* **44** (1972) 189–213.
- [46] C. G. Bollini and J. J. Giambiagi, “*Dimensional Renormalization: The Number of Dimensions as a Regularizing Parameter*,” *Nuovo Cim. B* **12** (1972) 20–26.
- [47] S. Weinzierl, *Feynman Integrals. A Comprehensive Treatment for Students and Researchers*. UNITEXT for Physics. Springer, 2022.

- [48] F. Jegerlehner, “*Facts of life with $\gamma(5)$* ,” *Eur. Phys. J. C* **18** (2001) 673–679.
- [49] H. Bélusca-Maïto, A. Ilakovac, P. Kühler, M. Mador-Božinović, D. Stöckinger, and M. Weißwange, “*Introduction to Renormalization Theory and Chiral Gauge Theories in Dimensional Regularization with Non-Anticommuting γ_5* ,” *Symmetry* **15** no. 3, (2023) 622.
- [50] S. A. Larin, “*The Renormalization of the axial anomaly in dimensional regularization*,” *Phys. Lett.* **B303** (1993) 113–118.
- [51] J. Korner, D. Kreimer, and K. Schilcher, “*A Practicable $\gamma(5)$ scheme in dimensional regularization*,” *Z. Phys. C* **54** (1992) 503–512.
- [52] T. Kinoshita, “*Mass singularities of Feynman amplitudes*,” *J. Math. Phys.* **3** (1962) 650–677.
- [53] T. D. Lee and M. Nauenberg, “*Degenerate Systems and Mass Singularities*,” *Phys. Rev.* **133** (1964) B1549–B1562.
- [54] F. Gross *et al.*, “*50 Years of Quantum Chromodynamics*,” *Eur. Phys. J. C* **83** (2023) 1125.
- [55] S. Catani and M. H. Seymour, “*A General algorithm for calculating jet cross-sections in NLO QCD*,” *Nucl. Phys. B* **485** (1997) 291–419. [Erratum: *Nucl. Phys. B* 510, 503–504 (1998)].
- [56] S. Frixione, Z. Kunszt, and A. Signer, “*Three jet cross-sections to next-to-leading order*,” *Nucl. Phys. B* **467** (1996) 399–442.
- [57] S. Frixione, “*A General approach to jet cross-sections in QCD*,” *Nucl. Phys. B* **507** (1997) 295–314.
- [58] R. Frederix, S. Frixione, F. Maltoni, and T. Stelzer, “*Automation of next-to-leading order computations in QCD: The FKS subtraction*,” *JHEP* **10** (2009) 003.
- [59] S. Catani, S. Dittmaier, M. H. Seymour, and Z. Trocsanyi, “*The Dipole formalism for next-to-leading order QCD calculations with massive partons*,” *Nucl. Phys. B* **627** (2002) 189–265.
- [60] A. Gehrmann-De Ridder, T. Gehrmann, and E. W. N. Glover, “*Antenna subtraction at NNLO*,” *JHEP* **09** (2005) 056.
- [61] J. Currie, E. W. N. Glover, and S. Wells, “*Infrared Structure at NNLO Using Antenna Subtraction*,” *JHEP* **04** (2013) 066.
- [62] S. Catani and M. Grazzini, “*An NNLO subtraction formalism in hadron collisions and its application to Higgs boson production at the LHC*,” *Phys. Rev. Lett.* **98** (2007) 222002.
- [63] S. Catani, L. Cieri, G. Ferrera, D. de Florian, and M. Grazzini, “*Vector boson production at hadron colliders: a fully exclusive QCD calculation at NNLO*,” *Phys. Rev. Lett.* **103** (2009) 082001.

- [64] I. W. Stewart, F. J. Tackmann, and W. J. Waalewijn, “*N-Jettiness: An Inclusive Event Shape to Veto Jets*,” *Phys. Rev. Lett.* **105** (2010) 092002.
- [65] R. Boughezal, C. Focke, X. Liu, and F. Petriello, “*W-boson production in association with a jet at next-to-next-to-leading order in perturbative QCD*,” *Phys. Rev. Lett.* **115** no. 6, (2015) 062002.
- [66] J. Gaunt, M. Stahlhofen, F. J. Tackmann, and J. R. Walsh, “*N-jettiness Subtractions for NNLO QCD Calculations*,” *JHEP* **09** (2015) 058.
- [67] M. Czakon, “*A novel subtraction scheme for double-real radiation at NNLO*,” *Phys. Lett. B* **693** (2010) 259–268.
- [68] F. Caola, K. Melnikov, and R. Röntsch, “*Nested soft-collinear subtractions in NNLO QCD computations*,” *Eur. Phys. J. C* **77** no. 4, (2017) 248.
- [69] G. Heinrich, “*Collider Physics at the Precision Frontier*,” *Phys. Rept.* **922** (2021) 1–69.
- [70] H. Cheng and T. T. Wu, *EXPANDING PROTONS: SCATTERING AT HIGH-ENERGIES*. MIT Press, Cambridge, 1987.
- [71] E. Panzer, *Feynman integrals and hyperlogarithms*. PhD thesis, Humboldt U., 2015.
- [72] R. N. Lee and A. A. Pomeransky, “*Critical points and number of master integrals*,” *JHEP* **11** (2013) 165.
- [73] V. A. Smirnov, *Analytic tools for Feynman integrals*, vol. 250. Springer, 2012.
- [74] E. Panzer, “*On hyperlogarithms and Feynman integrals with divergences and many scales*,” *JHEP* **03** (2014) 071.
- [75] A. von Manteuffel, E. Panzer, and R. M. Schabinger, “*A quasi-finite basis for multi-loop Feynman integrals*,” *JHEP* **02** (2015) 120.
- [76] L. D. Landau, “*On analytic properties of vertex parts in quantum field theory*,” *Nucl. Phys.* **13** no. 1, (1959) 181–192.
- [77] J. D. Bjorken, *Experimental tests of Quantum electrodynamics and spectral representations of Green’s functions in perturbation theory*. PhD thesis, Stanford U., 1959.
- [78] R. E. Cutkosky, “*Singularities and discontinuities of Feynman amplitudes*,” *J. Math. Phys.* **1** (1960) 429–433.
- [79] S. Coleman and R. E. Norton, “*Singularities in the physical region*,” *Nuovo Cim.* **38** (1965) 438–442.
- [80] R. J. Eden, P. V. Landshoff, D. I. Olive, and J. C. Polkinghorne, *The analytic S-matrix*. Cambridge Univ. Press, Cambridge, 1966.
- [81] E. Gardi, F. Herzog, S. Jones, and Y. Ma, “*Dissecting polytopes: Landau singularities and asymptotic expansions in $2 \rightarrow 2$ scattering*,” *JHEP* **08** (2024) 127.

- [82] D. E. Soper, “*Talk on QCD calculations by numerical integration*,” in *4th International Symposium on Radiative Corrections: Applications of Quantum Field Theory to Phenomenology*, pp. 305–315. 9, 1998.
- [83] D. E. Soper, “*Techniques for QCD calculations by numerical integration*,” *Phys. Rev. D* **62** (2000) 014009.
- [84] T. Binoth, G. Heinrich, and N. Kauer, “*A Numerical evaluation of the scalar hexagon integral in the physical region*,” *Nucl. Phys. B* **654** (2003) 277–300.
- [85] T. Binoth, J. P. Guillet, G. Heinrich, E. Pilon, and C. Schubert, “*An Algebraic/numerical formalism for one-loop multi-leg amplitudes*,” *JHEP* **10** (2005) 015.
- [86] Z. Nagy and D. E. Soper, “*Numerical integration of one-loop Feynman diagrams for N-photon amplitudes*,” *Phys. Rev. D* **74** (2006) 093006.
- [87] C. Anastasiou, S. Beerli, S. Bucherer, A. Daleo, and Z. Kunszt, “*Two-loop amplitudes and master integrals for the production of a Higgs boson via a massive quark and a scalar-quark loop*,” *JHEP* **01** (2007) 082.
- [88] C. Anastasiou, S. Beerli, and A. Daleo, “*Evaluating multi-loop Feynman diagrams with infrared and threshold singularities numerically*,” *JHEP* **05** (2007) 071.
- [89] A. Lazopoulos, K. Melnikov, and F. Petriello, “*QCD corrections to tri-boson production*,” *Phys. Rev. D* **76** (2007) 014001.
- [90] A. Lazopoulos, K. Melnikov, and F. J. Petriello, “*NLO QCD corrections to the production of $t\bar{t}Z$ in gluon fusion*,” *Phys. Rev. D* **77** (2008) 034021.
- [91] C. Anastasiou, S. Beerli, and A. Daleo, “*The Two-loop QCD amplitude $gg \rightarrow h, H$ in the Minimal Supersymmetric Standard Model*,” *Phys. Rev. Lett.* **100** (2008) 241806.
- [92] W. Gong, Z. Nagy, and D. E. Soper, “*Direct numerical integration of one-loop Feynman diagrams for N-photon amplitudes*,” *Phys. Rev. D* **79** (2009) 033005.
- [93] S. Becker, C. Reuschle, and S. Weinzierl, “*Numerical NLO QCD calculations*,” *JHEP* **12** (2010) 013.
- [94] S. C. Borowka, *Evaluation of multi-loop multi-scale integrals and phenomenological two-loop applications*. PhD thesis, Munich, Tech. U., 2014.
- [95] L. Chen, “*A prescription for projectors to compute helicity amplitudes in D dimensions*,” *Eur. Phys. J. C* **81** no. 5, (2021) 417.
- [96] T. Peraro and L. Tancredi, “*Physical projectors for multi-leg helicity amplitudes*,” *JHEP* **07** (2019) 114.
- [97] P. Nogueira, “*Automatic Feynman Graph Generation*,” *J. Comput. Phys.* **105** (1993) 279–289.
- [98] T. Hahn, “*Generating Feynman diagrams and amplitudes with FeynArts 3*,” *Comput. Phys. Commun.* **140** (2001) 418–431.

- [99] J. Braun, “*FeynGraph - A Modern High-Performance Feynman Diagram Generator*,” 2025. [Contribution at ACAT](#).
- [100] B. Ruijl, T. Ueda, and J. Vermaseren, “*FORM version 4.2*,”.
- [101] T. van Ritbergen, A. N. Schellekens, and J. A. M. Vermaseren, “*Group theory factors for Feynman diagrams*,” *Int. J. Mod. Phys. A* **14** (1999) 41–96.
- [102] V. Magerya, “Amplitude library (ALIBRARY): gluing all the tools needed for computing multi-loop amplitudes in QCD and beyond.”
<https://github.com/magv/alibrary>.
- [103] V. Shtabovenko, R. Mertig, and F. Orellana, “*New Developments in FeynCalc 9.0*,” *Comput. Phys. Commun.* **207** (2016) 432–444.
- [104] V. Shtabovenko, R. Mertig, and F. Orellana, “*FeynCalc 9.3: New features and improvements*,” *Comput. Phys. Commun.* **256** (2020) 107478.
- [105] V. Shtabovenko, R. Mertig, and F. Orellana, “*FeynCalc 10: Do multiloop integrals dream of computer codes?*,” *Comput. Phys. Commun.* **306** (2025) 109357.
- [106] M. Gerlach, F. Herren, and M. Lang, “*tapir: A tool for topologies, amplitudes, partial fraction decomposition and input for reductions*,” *Comput. Phys. Commun.* **282** (2023) 108544.
- [107] K. G. Chetyrkin and F. V. Tkachov, “*Integration by parts: The algorithm to calculate β -functions in 4 loops*,” *Nucl. Phys. B* **192** (1981) 159–204.
- [108] F. V. Tkachov, “*A theorem on analytical calculability of 4-loop renormalization group functions*,” *Phys. Lett. B* **100** (1981) 65–68.
- [109] D. Artico and L. Magnea, “*Integration-by-parts identities and differential equations for parametrised Feynman integrals*,” *JHEP* **03** (2024) 096.
- [110] G. ’t Hooft and M. J. G. Veltman, “*Scalar One Loop Integrals*,” *Nucl. Phys. B* **153** (1979) 365–401.
- [111] T. Gehrmann and E. Remiddi, “*Differential equations for two-loop four-point functions*,” *Nucl. Phys. B* **580** (2000) 485–518.
- [112] R. N. Lee, “*Group structure of the integration-by-part identities and its application to the reduction of multiloop integrals*,” *JHEP* **07** (2008) 031.
- [113] S. Laporta, “*High-precision calculation of multiloop Feynman integrals by difference equations*,” *Int. J. Mod. Phys. A* **15** (2000) 5087–5159.
- [114] F. Lange, J. Usovitsch, and Z. Wu, “*Kira 3: integral reduction with efficient seeding and optimized equation selection*,”.
- [115] A. V. Smirnov and F. S. Chukharev, “*FIRE6: Feynman Integral REduction with modular arithmetic*,” *Comput. Phys. Commun.* **247** (2020) 106877.
- [116] T. Peraro, “*FiniteFlow: multivariate functional reconstruction using finite fields and dataflow graphs*,” *JHEP* **07** (2019) 031.

- [117] J. Klappert and F. Lange, “*Reconstructing rational functions with FireFly*,” *Comput. Phys. Commun.* **247** (2020) 106951.
- [118] J. Klappert, S. Y. Klein, and F. Lange, “*Interpolation of dense and sparse rational functions and other improvements in FireFly*,” *Comput. Phys. Commun.* **264** (2021) 107968.
- [119] A. von Manteuffel and R. M. Schabinger, “*A novel approach to integration by parts reduction*,” *Phys. Lett. B* **744** (2015) 101–104.
- [120] X. Liu, “*Reconstruction of rational functions made simple*,” *Phys. Lett. B* **850** (2024) 138491.
- [121] P. Maierhöfer, J. Usovitsch, and P. Uwer, “*Kira—A Feynman integral reduction program*,” *Comput. Phys. Commun.* **230** (2018) 99–112.
- [122] J. Klappert, F. Lange, P. Maierhöfer, and J. Usovitsch, “*Integral reduction with Kira 2.0 and finite field methods*,” *Comput. Phys. Commun.* **266** (2021) 108024.
- [123] A. V. Smirnov and M. Zeng, “*FIRE 6.5: Feynman integral reduction with new simplification library*,” *Comput. Phys. Commun.* **302** (2024) 109261.
- [124] C. Studerus, “*Reduze – Feynman integral reduction in C++*,” *Comput. Phys. Commun.* **181** (2010) 1293–1300.
- [125] A. von Manteuffel and C. Studerus, “*Reduze 2 - Distributed Feynman Integral Reduction*,” <https://arxiv.org/abs/1201.4330>.
- [126] V. Magerya, “*Rational Tracer: a Tool for Faster Rational Function Reconstruction*,” <https://arxiv.org/abs/2211.03572>.
- [127] K. Melnikov, L. Tancredi, and C. Wever, “*Two-loop $gg \rightarrow Hg$ amplitude mediated by a nearly massless quark*,” *JHEP* **11** (2016) 104.
- [128] A. V. Smirnov and V. A. Smirnov, “*How to choose master integrals*,” *Nucl. Phys. B* **960** (2020) 115213.
- [129] J. Usovitsch, “*Factorization of denominators in integration-by-parts reductions*,” <https://arxiv.org/abs/2002.08173>.
- [130] J. P. Guillet, E. Pilon, Y. Shimizu, and M. S. Zidi, “*Framework for a novel mixed analytical/numerical approach for the computation of two-loop N -point Feynman diagrams*,” *PTEP* **2020** no. 4, (2020) 043B01.
- [131] A. V. Kotikov, “*Differential equations method: New technique for massive Feynman diagrams calculation*,” *Phys. Lett. B* **254** (1991) 158–164.
- [132] E. Remiddi, “*Differential equations for Feynman graph amplitudes*,” *Nuovo Cim. A* **110** (1997) 1435–1452.
- [133] J. M. Henn, “*Multiloop integrals in dimensional regularization made simple*,” *Phys. Rev. Lett.* **110** (2013) 251601.
- [134] A. B. Goncharov, “*Multiple polylogarithms, cyclotomy and modular complexes*,” *Math. Res. Lett.* **5** (1998) 497–516.

- [135] C. Duhr, “*Hopf algebras, coproducts and symbols: an application to Higgs boson amplitudes*,” *JHEP* **08** (2012) 043.
- [136] E. Remiddi and J. A. M. Vermaseren, “*Harmonic polylogarithms*,” *Int. J. Mod. Phys. A* **15** (2000) 725–754.
- [137] S. Pögel, X. Wang, and S. Weinzierl, “*Bananas of equal mass: any loop, any order in the dimensional regularisation parameter*,” *JHEP* **04** (2023) 117.
- [138] M. Czakon and T. Schutzmeier, “*Double fermionic contributions to the heavy-quark vacuum polarization*,” *JHEP* **07** (2008) 001.
- [139] M. Czakon, “*Tops from Light Quarks: Full Mass Dependence at Two-Loops in QCD*,” *Phys. Lett. B* **664** (2008) 307–314.
- [140] T. Binoth and G. Heinrich, “*An automatized algorithm to compute infrared divergent multi-loop integrals*,” *Nucl. Phys. B* **585** (2000) 741–759.
- [141] T. Binoth and G. Heinrich, “*Numerical evaluation of multi-loop integrals by sector decomposition*,” *Nucl. Phys. B* **680** (2004) 375–388.
- [142] G. Heinrich, “*Sector Decomposition*,” *Int. J. Mod. Phys. A* **23** (2008) 1457–1486.
- [143] A. Gehrmann-De Ridder, T. Gehrmann, and G. Heinrich, “*Four particle phase space integrals in massless QCD*,” *Nucl. Phys. B* **682** (2004) 265–288.
- [144] C. Anastasiou, K. Melnikov, and F. Petriello, “*A new method for real radiation at NNLO*,” *Phys. Rev. D* **69** (2004) 076010.
- [145] C. Anastasiou, K. Melnikov, and F. Petriello, “*Real radiation at NNLO: $e^+ e^- \rightarrow 2$ jets through $O(\alpha^2(s))$* ,” *Phys. Rev. Lett.* **93** (2004) 032002.
- [146] T. Binoth and G. Heinrich, “*Numerical evaluation of phase space integrals by sector decomposition*,” *Nucl. Phys. B* **693** (2004) 134–148.
- [147] G. Heinrich, “*Towards $e^+ e^- \rightarrow 3$ jets at NNLO by sector decomposition*,” *Eur. Phys. J. C* **48** (2006) 25–33.
- [148] E. de Doncker, Y. Shimizu, J. Fujimoto, and F. Yuasa, “*Computation of loop integrals using extrapolation*,” *Comput. Phys. Commun.* **159** (2004) 145–156.
- [149] F. Yuasa, E. de Doncker, N. Hamaguchi, T. Ishikawa, K. Kato, Y. Kurihara, J. Fujimoto, and Y. Shimizu, “*Numerical Computation of Two-loop Box Diagrams with Masses*,” *Comput. Phys. Commun.* **183** (2012) 2136–2144.
- [150] E. de Doncker, F. Yuasa, K. Kato, T. Ishikawa, J. Kapenga, and O. Olagbemi, “*Regularization with Numerical Extrapolation for Finite and UV-Divergent Multi-loop Integrals*,” *Comput. Phys. Commun.* **224** (2018) 164–185.
- [151] S. Catani, T. Gleisberg, F. Krauss, G. Rodrigo, and J.-C. Winter, “*From loops to trees by-passing Feynman’s theorem*,” *JHEP* **09** (2008) 065.
- [152] W. Kilian and T. Kleinschmidt, “*Numerical Evaluation of Feynman Loop Integrals by Reduction to Tree Graphs*,” <https://arxiv.org/abs/0912.3495>.
- [153] I. Bierenbaum, S. Catani, P. Draggiotis, and G. Rodrigo, “*A Tree-Loop Duality Relation at Two Loops and Beyond*,” *JHEP* **10** (2010) 073.

- [154] I. Bierenbaum, S. Buchta, P. Draggiotis, I. Malamos, and G. Rodrigo, “*Tree-Loop Duality Relation beyond simple poles*,” **JHEP** **03** (2013) 025.
- [155] S. Buchta, G. Chachamis, P. Draggiotis, and G. Rodrigo, “*Numerical implementation of the loop–tree duality method*,” **Eur. Phys. J. C** **77** no. 5, (2017) 274.
- [156] R. J. Hernandez-Pinto, G. F. R. Sborlini, and G. Rodrigo, “*Towards gauge theories in four dimensions*,” **JHEP** **02** (2016) 044.
- [157] G. F. R. Sborlini, F. Driencourt-Mangin, R. Hernandez-Pinto, and G. Rodrigo, “*Four-dimensional unsubtraction from the loop-tree duality*,” **JHEP** **08** (2016) 160.
- [158] M. Czakon, “*Automatized analytic continuation of Mellin-Barnes integrals*,” **Comput. Phys. Commun.** **175** (2006) 559–571.
- [159] J. Baglio, F. Campanario, S. Glaus, M. Mühlleitner, M. Spira, and J. Streicher, “*Gluon fusion into Higgs pairs at NLO QCD and the top mass scheme*,” **Eur. Phys. J. C** **79** no. 6, (2019) 459.
- [160] J. Baglio, F. Campanario, S. Glaus, M. Mühlleitner, J. Ronca, M. Spira, and J. Streicher, “*Higgs-Pair Production via Gluon Fusion at Hadron Colliders: NLO QCD Corrections*,” **JHEP** **04** (2020) 181.
- [161] L. F. Richardson and J. A. Gaunt, “*The deferred approach to the limit*,” **Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character** **226** (1927) 299–361.
- [162] J. de Jesús Aguilera-Verdugo *et al.*, “*A Stroll through the Loop-Tree Duality*,” **Symmetry** **13** no. 6, (2021) 1029.
- [163] A. Pak and A. Smirnov, “*Geometric approach to asymptotic expansion of Feynman integrals*,” **Eur. Phys. J. C** **71** (2011) 1626.
- [164] B. Ananthanarayan, A. Pal, S. Ramanan, and R. Sarkar, “*Unveiling Regions in multi-scale Feynman Integrals using Singularities and Power Geometry*,” **Eur. Phys. J. C** **79** no. 1, (2019) 57.
- [165] B. Ananthanarayan, A. B. Das, and R. Sarkar, “*Asymptotic analysis of Feynman diagrams and their maximal cuts*,” **Eur. Phys. J. C** **80** no. 12, (2020) 1131.
- [166] L. Bellafronte, G. Degrassi, P. P. Giardino, R. Gröber, and M. Vitti, “*Gluon fusion production at NLO: merging the transverse momentum and the high-energy expansions*,” **JHEP** **07** (2022) 069.
- [167] G. Degrassi, R. Gröber, M. Vitti, and X. Zhao, “*On the NLO QCD corrections to gluon-initiated ZH production*,” **JHEP** **08** (2022) 009.
- [168] J. Davies, G. Mishima, K. Schönwald, and M. Steinhauser, “*Analytic approximations of $2 \rightarrow 2$ processes with massive internal particles*,” **JHEP** **06** (2023) 063.
- [169] J. Davies, K. Schönwald, M. Steinhauser, and H. Zhang, “*Next-to-leading order electroweak corrections to $gg \rightarrow HH$ and $gg \rightarrow gH$ in the large- m_t limit*,” **JHEP** **10** (2023) 033.

- [170] M. Roth and A. Denner, “*High-energy approximation of one loop Feynman integrals*,” *Nucl. Phys. B* **479** (1996) 495–514.
- [171] A. Denner and S. Pozzorini, “*An Algorithm for the high-energy expansion of multi-loop diagrams to next-to-leading logarithmic accuracy*,” *Nucl. Phys. B* **717** (2005) 48–85.
- [172] S. Pozzorini, “*Next to leading mass singularities in two loop electroweak singlet form-factors*,” *Nucl. Phys. B* **692** (2004) 135–174.
- [173] C. Bogner and S. Weinzierl, “*Resolution of singularities for multi-loop integrals*,” *Comput. Phys. Commun.* **178** (2008) 596–610.
- [174] H. Hironaka, “*Resolution of Singularities of an Algebraic Variety Over a Field of Characteristic Zero: I*,” *Annals of Mathematics* **79** no. 1, (1964) 109–203.
<http://www.jstor.org/stable/1970486>.
- [175] T. Ueda and J. Fujimoto, “*New implementation of the sector decomposition on FORM*,” *PoS ACAT08* (2008) 120.
- [176] T. Kaneko and T. Ueda, “*A Geometric method of sector decomposition*,” *Comput. Phys. Commun.* **181** (2010) 1352–1361.
- [177] T. Kaneko and T. Ueda, “*Sector Decomposition Via Computational Geometry*,” *PoS ACAT2010* (2010) 082.
- [178] S. Borowka, G. Heinrich, S. P. Jones, M. Kerner, J. Schlenk, and T. Zirke, “*SecDec-3.0: Numerical evaluation of multi-scale integrals beyond one loop*,” *Comput. Phys. Commun.* **196** (2015) 470–491.
- [179] J. K. Schlenk, *Techniques for higher order corrections and their application to LHC phenomenology*. PhD thesis, Munich, Tech. U., 8, 2016.
- [180] S. Borowka, G. Heinrich, S. Jahn, S. P. Jones, M. Kerner, J. Schlenk, and T. Zirke, “*pySecDec: a toolbox for the numerical evaluation of multi-scale integrals*,” *Comput. Phys. Commun.* **222** (2018) 313–326.
- [181] S. Borowka, G. Heinrich, S. Jahn, S. P. Jones, M. Kerner, and J. Schlenk, “*A GPU compatible quasi-Monte Carlo integrator interfaced to pySecDec*,” *Comput. Phys. Commun.* **240** (2019) 120–137.
- [182] S. Jahn, *Automation of Multi-Loop Amplitude Calculations*. PhD thesis, Munich, Tech. U., 2020.
- [183] G. Heinrich, S. Jahn, S. P. Jones, M. Kerner, F. Langer, V. Magerya, A. Pöldaru, J. Schlenk, and E. Villa, “*Expansion by regions with pySecDec*,” *Comput. Phys. Commun.* **273** (2022) 108267.
- [184] J. Dick, F. Y. Kuo, and I. H. Sloan, “*High-dimensional integration: The quasi-Monte Carlo way*,” *Acta Numerica* **22** (2013) 133–288.
- [185] T. Goda and P. L’Ecuyer, “*Construction-Free Median Quasi-Monte Carlo Rules for Function Spaces with Unspecified Smoothness and General Weights*,” *SIAM Journal on Scientific Computing* **44** no. 4, (Aug., 2022) A2765–A2788.
<http://dx.doi.org/10.1137/22M1473625>.

- [186] R. Narayanan and H. Neuberger, “*Infinite N phase transitions in continuum Wilson loop operators*,” **JHEP** **03** (2006) 064.
- [187] M. Luscher, “*Trivializing maps, the Wilson flow and the HMC algorithm*,” **Commun. Math. Phys.** **293** (2010) 899–919.
- [188] M. Lüscher, “*Properties and uses of the Wilson flow in lattice QCD*,” **JHEP** **08** (2010) 071. [Erratum: JHEP 03, 092 (2014)].
- [189] M. Luscher and P. Weisz, “*Perturbative analysis of the gradient flow in non-abelian gauge theories*,” **JHEP** **02** (2011) 051.
- [190] M. Luscher, “*Chiral symmetry and the Yang–Mills gradient flow*,” **JHEP** **04** (2013) 123.
- [191] R. V. Harlander, Y. Kluth, and F. Lange, “*The two-loop energy–momentum tensor within the gradient-flow formalism*,” **Eur. Phys. J. C** **78** no. 11, (2018) 944. [Erratum: Eur.Phys.J.C 79, 858 (2019)].
- [192] R. V. Harlander and T. Neumann, “*The perturbative QCD gradient flow to three loops*,” **JHEP** **06** (2016) 161.
- [193] J. Artz, R. V. Harlander, F. Lange, T. Neumann, and M. Prausa, “*Results and techniques for higher order calculations within the gradient-flow formalism*,” **JHEP** **06** (2019) 121. [Erratum: JHEP 10, 032 (2019)].
- [194] G. E. Collins, “*Quantifier elimination for real closed fields by cylindrical algebraic decomposition*,” in *Automata Theory and Formal Languages*, H. Brakhage, ed., pp. 134–183. Springer Berlin Heidelberg, Berlin, Heidelberg, 1975.
- [195] Z. Bern, L. J. Dixon, and D. A. Kosower, “*Dimensionally regulated pentagon integrals*,” **Nucl. Phys. B** **412** (1994) 751–816.
- [196] T. Binoth, J. P. Guillet, and G. Heinrich, “*Reduction formalism for dimensionally regulated one loop N point integrals*,” **Nucl. Phys. B** **572** (2000) 361–386.
- [197] G. Duplancic and B. Nizic, “*Reduction method for dimensionally regulated one loop N point Feynman integrals*,” **Eur. Phys. J. C** **35** (2004) 105–118.
- [198] W. T. Giele and E. W. N. Glover, “*A Computational formalism for one loop integrals*,” **JHEP** **04** (2004) 029.
- [199] S. Jones, A. Olsson, and T. Stone, “*Evaluating Parametric Integrals in the Minkowski Regime without Contour Deformation*,” **PoS LL2024** (2024) 036.
- [200] D. J. Broadhurst, J. Fleischer, and O. V. Tarasov, “*Two loop two point functions with masses: Asymptotic expansions and Taylor series, in any dimension*,” **Z. Phys. C** **60** (1993) 287–302.
- [201] S. Bloch and P. Vanhove, “*The elliptic dilogarithm for the sunset graph*,” **J. Number Theor.** **148** (2015) 328–364.
- [202] L. Adams, C. Bogner, and S. Weinzierl, “*The two-loop sunrise integral around four space-time dimensions and generalisations of the Clausen and Glaisher functions towards the elliptic case*,” **J. Math. Phys.** **56** no. 7, (2015) 072303.

- [203] E. Remiddi and L. Tancredi, “*An Elliptic Generalization of Multiple Polylogarithms*,” *Nucl. Phys. B* **925** (2017) 212–251.
- [204] J. B. Tausk, “*Nonplanar massless two loop Feynman diagrams with four on-shell legs*,” *Phys. Lett. B* **469** (1999) 225–234.
- [205] M. Borinsky, “*Tropical Monte Carlo quadrature for Feynman integrals*,” *Ann. Inst. H. Poincaré D Comb. Phys. Interact.* **10** no. 4, (2023) 635–685.
- [206] M. Borinsky, H. J. Munch, and F. Tellander, “*Tropical Feynman integration in the Minkowski regime*,” *Comput. Phys. Commun.* **292** (2023) 108874.
- [207] D. Maître and R. Santos-Mateos, “*Multi-variable integration with a neural network*,” *JHEP* **03** (2023) 221.
- [208] M. Zeng, “*Feynman integrals from positivity constraints*,” *JHEP* **09** (2023) 042.
- [209] T. Bitoun, C. Bogner, R. P. Klausen, and E. Panzer, “*Feynman integral relations from parametric annihilators*,” *Lett. Math. Phys.* **109** no. 3, (2019) 497–564.
- [210] W. Chen, “*Reduction of Feynman Integrals in the Parametric Representation*,” *JHEP* **02** (2020) 115.
- [211] W. Chen, “*Reduction of Feynman Integrals in the Parametric Representation II: Reduction of Tensor Integrals*,” *Eur. Phys. J. C* **81** no. 3, (2021) 244.
- [212] W. Chen, “*Reduction of Feynman integrals in the parametric representation III: integrals with cuts*,” *Eur. Phys. J. C* **80** no. 12, (2020) 1173.
- [213] S. Borowka, T. Gehrmann, and D. Hulme, “*Systematic approximation of multi-scale Feynman integrals*,” *JHEP* **08** (2018) 111.
- [214] H. S. Hannesdottir and S. Mizera, *What is the $i\epsilon$ for the S-matrix?* SpringerBriefs in Physics. Springer, 1, 2023.
- [215] J. Henn and P. Raman, “*Positivity properties of scattering amplitudes*,” *JHEP* **04** (2025) 150.
- [216] E. Herrmann and J. Trnka, “*The SAGEX review on scattering amplitudes Chapter 7: Positive geometry of scattering amplitudes*,” *J. Phys. A* **55** no. 44, (2022) 443008.
- [217] R. Britto, “*Generalized Cuts of Feynman Integrals in Parameter Space*,” *Phys. Rev. Lett.* **131** no. 9, (2023) 091601.
- [218] B. Jantzen, “*Foundation and generalization of the expansion by regions*,” *JHEP* **12** (2011) 076.
- [219] B. Jantzen, A. V. Smirnov, and V. A. Smirnov, “*Expansion by regions: revealing potential and Glauber regions automatically*,” *Eur. Phys. J. C* **72** (2012) 2139.
- [220] NNLOJET Collaboration, A. Huss *et al.*, “*NNLOJET: a parton-level event generator for jet cross sections at NNLO QCD accuracy*,”.
- [221] M. Grazzini, S. Kallweit, and M. Wiesemann, “*Fully differential NNLO computations with MATRIX*,” *Eur. Phys. J. C* **78** no. 7, (2018) 537.

- [222] J. Campbell and T. Neumann, “*Precision Phenomenology with MCFM*,” **JHEP** **12** (2019) 034.
- [223] S. Devoto, M. Grazzini, S. Kallweit, J. Mazzitelli, and C. Savoini, “*Precise predictions for $t\bar{t}H$ production at the LHC: inclusive cross section and differential distributions*,” **JHEP** **03** (2025) 189.
- [224] J. Davies, D. Grau, K. Schönwald, M. Steinhauser, D. Stremmer, and M. Vitti, “*Two-loop QCD corrections to ZH and off-shell Z boson pair production in gluon fusion*,”
- [225] M. Czakon, P. Fiedler, and A. Mitov, “*Total Top-Quark Pair-Production Cross Section at Hadron Colliders Through $O(\alpha_s^4)$* ,” **Phys. Rev. Lett.** **110** (2013) 252004.
- [226] L. Chen, M. Czakon, and R. Poncelet, “*Polarized double-virtual amplitudes for heavy-quark pair production*,” **JHEP** **03** (2018) 085.
- [227] S. Borowka, N. Greiner, G. Heinrich, S. P. Jones, M. Kerner, J. Schlenk, U. Schubert, and T. Zirke, “*Higgs Boson Pair Production in Gluon Fusion at Next-to-Leading Order with Full Top-Quark Mass Dependence*,” **Phys. Rev. Lett.** **117** no. 1, (2016) 012001. [Erratum: Phys.Rev.Lett. 117, 079901 (2016)].
- [228] G. Heinrich, S. P. Jones, M. Kerner, G. Luisoni, and E. Vryonidou, “*NLO predictions for Higgs boson pair production with full top quark mass dependence matched to parton showers*,” **JHEP** **08** (2017) 088.
- [229] S. P. Jones, M. Kerner, and G. Luisoni, “*Next-to-Leading-Order QCD Corrections to Higgs Boson Plus Jet Production with Full Top-Quark Mass Dependence*,” **Phys. Rev. Lett.** **120** no. 16, (2018) 162001. [Erratum: Phys.Rev.Lett. 128, 059901 (2022)].
- [230] B. Campillo Azeleira, G. Heinrich, M. Kerner, and L. Kunz, “*Probing anomalous Higgs boson couplings in Higgs plus jet production at NLO QCD with full m_t -dependence*,” **JHEP** **04** (2025) 060.
- [231] S. Alekhin, J. Blümlein, S. Moch, and R. Placakyte, “*Parton distribution functions, α_s , and heavy-quark masses for LHC Run II*,” **Phys. Rev. D** **96** no. 1, (2017) 014011.
- [232] **GoSam** Collaboration, G. Cullen *et al.*, “*GoSAM-2.0: a tool for automated one-loop calculations within the Standard Model and beyond*,” **Eur. Phys. J. C** **74** no. 8, (2014) 3001.
- [233] R. Kleiss, W. J. Stirling, and S. D. Ellis, “*A New Monte Carlo Treatment of Multiparticle Phase Space at High-energies*,” **Comput. Phys. Commun.** **40** (1986) 359.
- [234] G. P. Lepage, “*A new algorithm for adaptive multidimensional integration*,” **J. Comput. Phys.** **27** no. 2, (1978) 192–203.
- [235] I. Kobyzev, S. J. Prince, and M. A. Brubaker, “*Normalizing Flows: An Introduction and Review of Current Methods*,” **IEEE Transactions on Pattern Analysis and Machine Intelligence** **43** no. 11, (2021) 3964–3979.

- [236] T. Heimel, R. Winterhalder, A. Butter, J. Isaacson, C. Krause, F. Maltoni, O. Mattelaer, and T. Plehn, “*MadNIS - Neural multi-channel importance sampling*,” *SciPost Phys.* **15** no. 4, (2023) 141.
- [237] T. Heimel, N. Huetsch, F. Maltoni, O. Mattelaer, T. Plehn, and R. Winterhalder, “*The MadNIS reloaded*,” *SciPost Phys.* **17** no. 1, (2024) 023.
- [238] E. Bothmann, T. Janßen, M. Knobbe, T. Schmale, and S. Schumann, “*Exploring phase space with Neural Importance Sampling*,” *SciPost Phys.* **8** no. 4, (2020) 069.
- [239] E. Bothmann, T. Childers, W. Giele, F. Herren, S. Hoeche, J. Isaacson, M. Knobbe, and R. Wang, “*Efficient phase-space generation for hadron collider event simulation*,” *SciPost Phys.* **15** no. 4, (2023) 169.
- [240] N. Deutschmann and N. Götz, “*Accelerating HEP simulations with Neural Importance Sampling*,” *JHEP* **03** (2024) 083.
- [241] P. J. Davis, *Interpolation and Approximation*. Dover Publications, New York, 1975.
- [242] L. N. Trefethen, *Approximation Theory and Approximation Practice, Extended Edition*. SIAM, Philadelphia, PA, 2019.
- [243] J.-P. Berrut and L. N. Trefethen, “*Barycentric Lagrange Interpolation*,” *SIAM Rev.* **46** no. 3, (2004) 501–517.
- [244] Y. Nakatsukasa, O. Sete, and L. N. Trefethen, “*The first five years of the AAA algorithm*,” <https://arxiv.org/abs/2312.03565>.
- [245] C. Runge, “*Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten*,” *Z. Math. Phys.* **46** (1901) 224–243.
- [246] L. N. Trefethen, “*Six Myths of Polynomial Interpolation and Quadrature*,” *Math. Today* **47** no. 1, (2011) 184–188.
<https://people.maths.ox.ac.uk/trefethen/mythspaper.pdf>.
- [247] L. N. Trefethen, “*Exactness of Quadrature Formulas*,” *SIAM Rev.* **64** no. 1, (2022) 132–150.
- [248] H. Majidian, “*On the decay rate of Chebyshev coefficients*,” *Appl. Numer. Math.* **113** (2017) 44–53.
- [249] S. Xiang, “*On the optimal convergence rates of Chebyshev interpolations for functions of limited regularity*,” *Appl. Math. Lett.* **84** (2018) 1–7.
- [250] K. Glau and M. Mahlstedt, “*Improved error bound for multivariate Chebyshev polynomial interpolation*,” *Int. J. Comput. Math.* **96** no. 11, (2019) 2302–2314.
- [251] R. BELLMAN, *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961. <http://www.jstor.org/stable/j.ctt183ph6v>.
- [252] L. N. Trefethen, “*Multivariate polynomial approximation in the hypercube*,” *Proc. Amer. Math. Soc.* **145** no. 11, (2017) 4837–4844.

- [253] D. Dũng, V. N. Temlyakov, and T. Ullrich, *Hyperbolic Cross Approximation*. Birkhäuser Cham, 2018.
- [254] N. Dyn and M. S. Floater, “Multivariate polynomial interpolation on lower sets,” *J. Approx. Theory* **177** (2014) 34–42.
- [255] M. Griebel, M. Schneider, and C. Zenger, “A combination technique for the solution of sparse grid problems,” in *Iterative Methods in Linear Algebra*, P. de Groen and R. Beauwens, eds., pp. 263–281. Elsevier, 1992.
<https://api.semanticscholar.org/CorpusID:16460274>.
- [256] M. Hegland, “Adaptive sparse grids,” in *Proc. of 10th Computational Techniques and Applications Conference CTAC-2001*, K. Burrage and R. B. Sidje, eds., vol. 44, pp. C335–C353. Apr., 2003.
- [257] B. Adcock, S. Brugiapaglia, and C. G. Webster, *Sparse Polynomial Approximation of High-Dimensional Functions*. SIAM, Philadelphia, PA, 2022.
- [258] C. R. Morrow and T. N. L. Patterson, “Construction of Algebraic Cubature Rules Using Polynomial Ideal Theory,” *SIAM J. Numer. Anal.* **15** no. 5, (1978) 953–976.
- [259] Y. Xu, “Lagrange Interpolation on Chebyshev Points of Two Variables,” *J. Approx. Theory* **87** no. 2, (1996) 220–238.
- [260] M. Caliari, S. De Marchi, and M. Vianello, “Bivariate polynomial interpolation on the square at new nodal sets,” *Appl. Math. Comput.* **165** no. 2, (2005) 261–274.
- [261] L. Bos, M. Caliari, S. De Marchi, M. Vianello, and Y. Xu, “Bivariate Lagrange interpolation at the Padua points: The generating curve approach,” *J. Approx. Theory* **143** no. 1, (2006) 15–25. Special Issue on Foundations of Computational Mathematics.
- [262] R. Cools and K. Poppe, “Chebyshev lattices, a unifying framework for cubature with Chebyshev weight function,” *BIT Numer. Math.* **51** no. 2, (Jun, 2011) 275–288.
- [263] C. Kaethner, M. Ahlborg, P. Dencker, and W. Erb, “A survey on bivariate Lagrange interpolation on Lissajous nodes,” *Dolomites Res. Notes Approx.* **8** (2015) 23–36.
- [264] P. Dencker and W. Erb, “A unifying theory for multivariate polynomial interpolation on general Lissajous–Chebyshev nodes,”
<https://arxiv.org/abs/1711.00557>.
- [265] S. De Marchi, “On Leja sequences: some results and applications,” *Appl. Math. Comput.* **152** no. 3, (2004) 621–647.
- [266] M. Hecht, P.-A. Hofmann, D. Wicaksono, U. H. Acosta, K. Gonciarz, J. Kissinger, V. Sivkin, and I. F. Sbalzarini, “Multivariate Newton Interpolation in Downward Closed Spaces Reaches the Optimal Geometric Approximation Rates for Bos–Levenberg–Trefethen Functions,”

- [267] D. Wicaksono, U. H. Acosta, S. K. T. Veettil, J. Kissinger, and M. Hecht, “MINTERPY: *multivariate polynomial interpolation in Python*,” *J. Open Source Softw.* **10** no. 109, (2025) 7702.
<https://github.com/minterpy-project/minterpy>.
- [268] I. J. Schoenberg, “*Contributions to the problem of approximation of equidistant data by analytic functions. Part B. On the problem of osculatory interpolation. A second class of analytic approximation formulae*,” *Quart. Appl. Math.* **4** (1946) 112–141.
- [269] I. J. Schoenberg, “*Cardinal interpolation and spline functions*,” *J. Approx. Theory* **2** no. 2, (1969) 167–206.
- [270] K. Höllig, *Finite Element Methods with B-Splines*. Society for Industrial and Applied Mathematics, 2003.
- [271] K. Höllig and J. Hörner, *Approximation and Modeling with B-Splines*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2013.
- [272] G. Heinrich, S. Jones, M. Kerner, G. Luisoni, and E. Vryonidou, “*NLO predictions for Higgs boson pair production with full top quark mass dependence matched to parton showers*,” *JHEP* **2017** no. 8, (Aug., 2017) .
- [273] M. Czakon, F. Eschment, M. Niggetiedt, R. Poncelet, and T. Schellenberger, “*Quark mass effects in Higgs production*,” *JHEP* **2024** no. 10, (Oct., 2024) .
- [274] T. Carli, G. P. Salam, and F. Siegert, “*A posteriori inclusion of PDFs in NLO QCD final-state calculations*,” <https://arxiv.org/abs/hep-ph/0510324>.
- [275] M. G. Cox, “*The Numerical Evaluation of B-Splines*,” *IMA J. Appl. Math.* **10** no. 2, (10, 1972) 134–149.
- [276] C. de Boor, “*On calculating with B-Splines*,” *J. Approx. Theory* **6** no. 1, (1972) 50–62.
- [277] C.-K. Shene, “CS3621 Introduction to Computing with Geometry Notes.” <https://pages.mtu.edu/~shene/COURSES/cs3621/NOTES/>. Accessed: 2024-04-24.
- [278] T. Lyche, C. Manni, and H. Speleers, *Foundations of Spline Theory: B-Splines, Spline Approximation, and Hierarchical Refinement*, pp. 1–76. Springer, Cham, 2018.
- [279] K. Höllig and J. Hörner, *Chapter 4: B-Splines*, pp. 51–75. SIAM, Philadelphia, PA, 2013.
- [280] G. Kermarrec, V. Skytt, and T. Dokken, *Locally Refined B-Splines*, pp. 13–21. Springer, Cham, 2023.
- [281] S. A. Smolyak, “*Quadrature and interpolation formulas for tensor products of certain classes of functions*,” *Soviet Math. Dokl.* **4** (1963) 240–243.
- [282] C. Zenger, “*Sparse Grids*,” in *Parallel Algorithms for Partial Differential Equations*, W. Hackbusch, ed., vol. 31, pp. 241–251. Vieweg, 1991.

- [283] M. Griebel, “*A parallelizable and vectorizable multi-level algorithm on sparse grids*,” in *Parallel Algorithms for Partial Differential Equations*, W. Hackbusch, ed., vol. 31, pp. 94–100. Vieweg, 1991.
<https://api.semanticscholar.org/CorpusID:266150335>.
- [284] H.-J. Bungartz and M. Griebel, “*Sparse grids*,” *Acta Numerica* **13** (2004) 147–269.
- [285] J. Garcke, “*Sparse Grids in a Nutshell*,” in *Sparse Grids and Applications*, J. Garcke and M. Griebel, eds., pp. 57–80. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
- [286] D. Pflüger, *Spatially Adaptive Sparse Grids for High-Dimensional Problems*. PhD thesis, Technische Universität München., Feb., 2010.
<http://www.dr.hut-verlag.de/978-3-86853-555-6.html>.
- [287] J. Valentin and D. Pflüger, “*Hierarchical Gradient-Based Optimization with B-Splines on Sparse Grids*,” in *Sparse Grids and Applications - Stuttgart 2014*, J. Garcke and D. Pflüger, eds., pp. 315–336. Springer International Publishing, Cham, 2016.
- [288] J. Brumm and S. Scheidegger, “*Using Adaptive Sparse Grids to Solve High-Dimensional Dynamic Models*,” *Econometrica* **85** (09, 2017) 1575–1612.
- [289] J. Valentin, *B-Splines for Sparse Grids: Algorithms and Application to Higher-Dimensional Optimization*. PhD thesis, Universität Stuttgart, 2019.
- [290] R. F. Michael, *B-Splines on Sparse Grids for Uncertainty Quantification*. PhD thesis, Universität Stuttgart, 2021.
- [291] D. Pflüger, “*Spatially Adaptive Refinement*,” in *Sparse Grids and Applications*, J. Garcke and M. Griebel, eds., Lecture Notes in Computational Science and Engineering, pp. 243–262. Springer, Berlin Heidelberg, Oct., 2012.
- [292] H. Bungartz, “*Higher Order Finite Elements on Sparse Grids*,” 1995.
<https://api.semanticscholar.org/CorpusID:43922240>.
- [293] M. F. Rehme, S. Zimmer, and D. Pflüger, “*Hierarchical Extended B-splines for Approximations on Sparse Grids*,” in *Sparse Grids and Applications - Munich 2018*, H.-J. Bungartz, J. Garcke, and D. Pflüger, eds., pp. 187–203. Springer, Cham, 2021.
- [294] J. Aylett-Bullock, S. Badger, and R. Moodie, “*Optimising simulations for diphoton production at hadron colliders using amplitude neural networks*,” *JHEP* **08** (2021) 066.
- [295] D. Maître and H. Truong, “*A factorisation-aware Matrix element emulator*,” *JHEP* **11** (2021) 066.
- [296] S. Badger, A. Butter, M. Luchmann, S. Pitz, and T. Plehn, “*Loop amplitudes from precision networks*,” *SciPost Phys. Core* **6** (2023) 034.
- [297] D. Maître and H. Truong, “*One-loop matrix element emulation with factorisation awareness*,” *JHEP* **05** (2023) 159.

- [298] J. Spinner, V. Bresó, P. de Haan, T. Plehn, J. Thaler, and J. Brehmer, “Lorentz-Equivariant Geometric Algebra Transformers for High-Energy Physics,” <https://arxiv.org/abs/2405.14806>.
- [299] J. Brehmer, V. Bresó, P. de Haan, T. Plehn, H. Qu, J. Spinner, and J. Thaler, “A Lorentz-Equivariant Transformer for All of the LHC,” <https://arxiv.org/abs/2411.00446>.
- [300] F. Rosenblatt, *The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain*, vol. 65. American Psychological Association, 1958.
- [301] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning Representations by Back-Propagating Errors,” *Nature* **323** (1986) 533–536.
- [302] D. P. Kingma and J. Ba, “ADAM: A Method for Stochastic Optimization,” <https://arxiv.org/abs/1412.6980>.
- [303] I. Loshchilov and F. Hutter, “SGDR: Stochastic Gradient Descent with Restarts,” CoRR (2016) . <https://arxiv.org/abs/1608.03983>.
- [304] D. Hendrycks and K. Gimpel, “Gaussian Error Linear Units (GELUs),” <https://arxiv.org/abs/1606.08415>.
- [305] S. Gong, Q. Meng, J. Zhang, H. Qu, C. Li, S. Qian, W. Du, Z.-M. Ma, and T.-Y. Liu, “An efficient Lorentz equivariant graph neural network for jet tagging,” *JHEP* **07** (2022) 030.
- [306] D. Ruhe, J. Brandstetter, and P. Forré, “Clifford Group Equivariant Neural Networks,” <https://arxiv.org/abs/2305.11141>.
- [307] A. Bogatskiy, T. Hoffman, D. W. Miller, J. T. Offermann, and X. Liu, “Explainable equivariant neural networks for particle physics: PELICAN,” *JHEP* **03** (2024) 113.
- [308] D. Hestenes, *Space-time Algebra*. Documents on modern physics. Gordon and Breach, 1966. <https://books.google.de/books?id=0oRmatRYcs4C>.
- [309] J. Brehmer, P. de Haan, S. Behrends, and T. Cohen, “Geometric Algebra Transformer,” in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds., vol. 37. 2023.
- [310] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention Is All You Need,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds., vol. 30. Curran Associates, Inc., 2017.
- [311] R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, C. Xing, H. Zhang, Y. Lan, L. Wang, and T. Liu, “On Layer Normalization in the Transformer Architecture,” in *International Conference on Machine Learning*, pp. 10524–10533, PMLR. 2020.

- [312] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, “*Scaling laws for neural language models*,” 2020. <https://arxiv.org/abs/2001.08361>.
- [313] U. Sharma and J. Kaplan, “*A neural scaling law from the dimension of the data manifold*,”. <https://arxiv.org/abs/2004.10802>.
- [314] U. Sharma and J. Kaplan, “*Scaling Laws from the Data Manifold Dimension*,” *Journal of Machine Learning Research* **23** no. 9, (2022) 1–34.
- [315] Y. Bahri, E. Dyer, J. Kaplan, J. Lee, and U. Sharma, “*Explaining neural scaling laws*,” *Proceedings of the National Academy of Sciences* **121** no. 27, (2024) .
- [316] J. Batson and Y. Kahn, “*Scaling Laws in Jet Classification*,” 2023. <https://arxiv.org/abs/2312.02264>.
- [317] D. J. C. Mackay, “*Probable networks and plausible predictions — a review of practical Bayesian methods for supervised neural networks*,” *Netw.: Comput. Neural Syst* **6** no. 3, (1995) 469–505.
- [318] R. M. Neal, *Bayesian learning for neural networks*. PhD thesis, Toronto, 1995.
- [319] Y. Gal, *Uncertainty in Deep Learning*. PhD thesis, Cambridge, 2016.
- [320] A. Butter, N. Huetsch, S. Palacios Schweitzer, T. Plehn, P. Sorrenson, and J. Spinner, “*Jet diffusion versus JetGPT – Modern networks for the LHC*,” *SciPost Phys. Core* **8** (2025) 026.
- [321] F. D’Angelo and V. Fortuin, “*Repulsive Deep Ensembles are Bayesian*,” 2023. <https://arxiv.org/abs/2106.11642>.
- [322] H. Bahl, V. Bresó, G. De Crescenzo, and T. Plehn, “*Advancing Tools for Simulation-Based Inference*,”. <https://arxiv.org/abs/2410.07315>.
- [323] W. Beenakker, S. Dittmaier, M. Krämer, B. Plümper, M. Spira, and P. Zerwas, “*NLO QCD corrections to $t\bar{t}H$ production in hadron collisions*,” *Nucl. Phys. B* **653** (2003) 151–203.